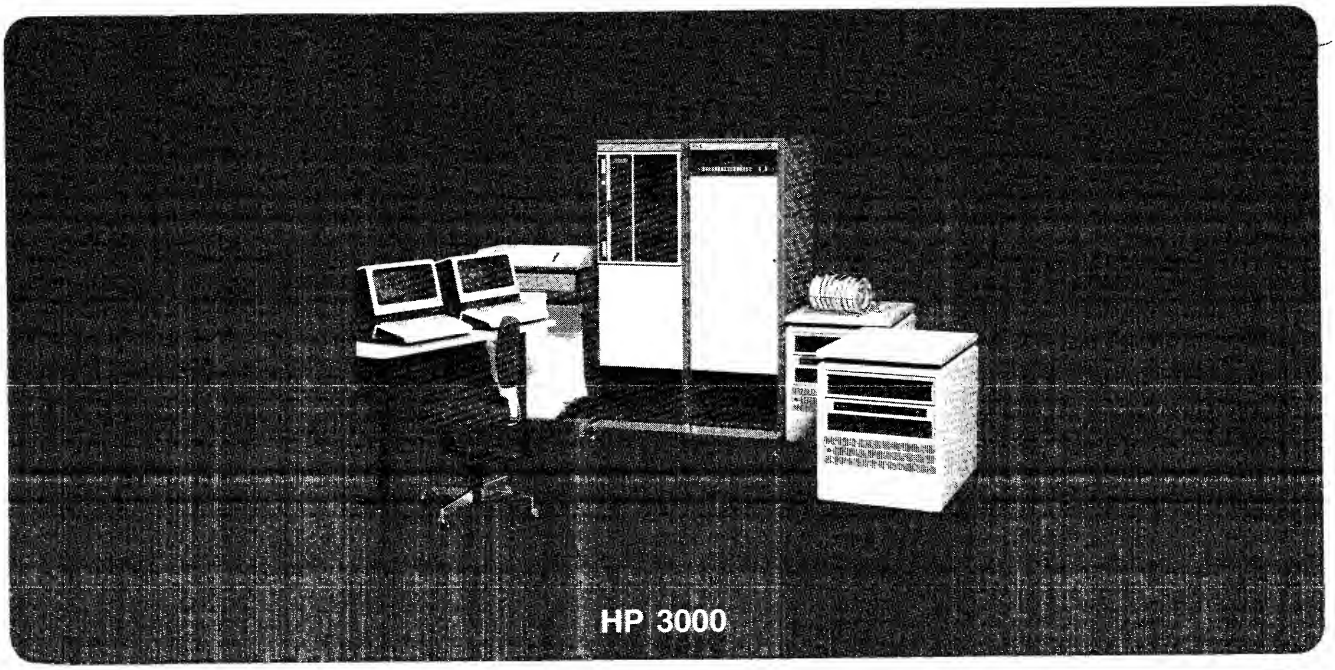
12/76

HEWLETT hp PACKARD

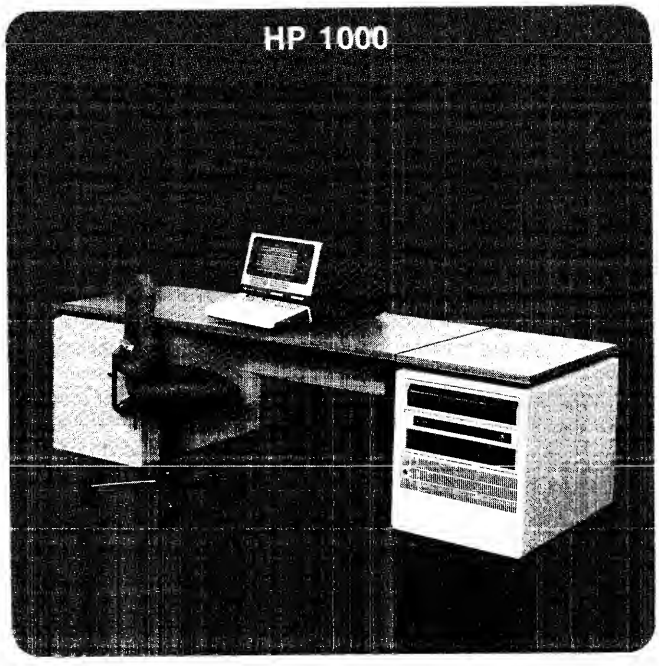
computer systems

COMMUNICATOR



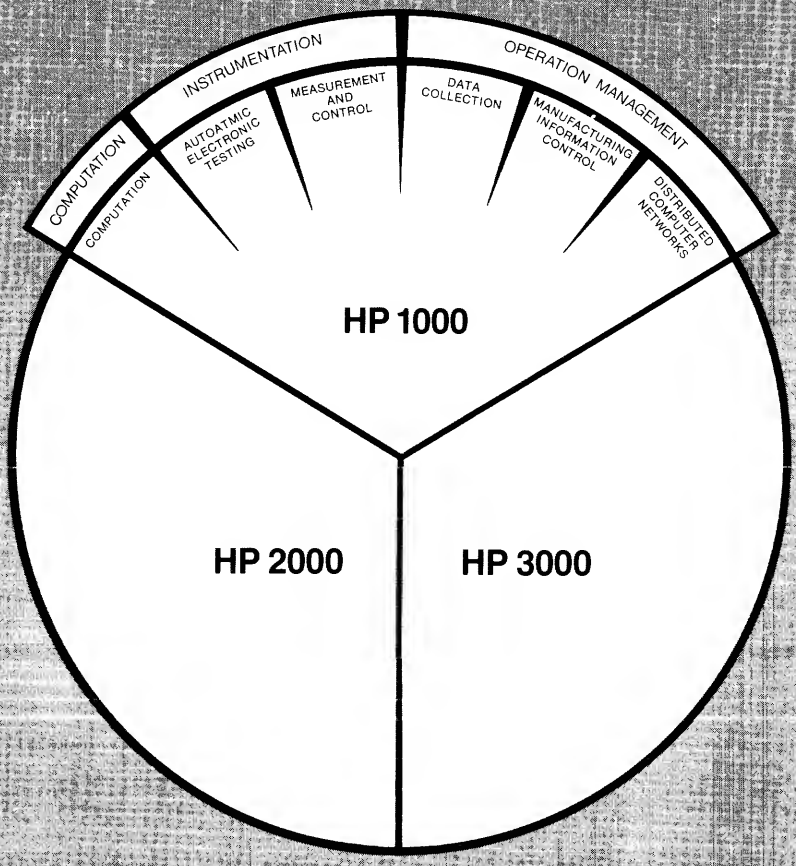
HP 3000

HP 2000



HP 1000

HP1000 Computer Systems



editor's note

This issue of the **COMMUNICATOR** features the new Hewlett-Packard 1000 Computer System introduced in October, 1976. This fast and powerful small computer system provides new user capabilities and incorporates recent computer advances setting a new price/performance standard in its class. The feature article introduces the HP 1000, outlining the Data Base Management capabilities, the key features, and the wide range of applications.

Concurrent with the introduction of the HP 1000, the **COMMUNICATOR** will now contain a section titled *ABOUT THE HP 1000*. This replaces the 9600/9700 title, reflecting an increase in the scope of HP Systems covered in this section.

The preparation of this issue of the **COMMUNICATOR** also coincides with the introduction in October, 1976 of revised software for the HP 2000 Operating System. New system capabilities provided by the software revision are summarized in the section *ABOUT THE HP 2000*.

The *3000 Bulletin* section describes the restructuring of General Systems Division training centers and the announcement of a Phone-In Consulting Service. These bulletins, of interest to all HP 3000 users, will describe the expanded role of support services.

With the start of the new fiscal and calendar year, General Systems Division of Hewlett-Packard is reviewing the contents and purpose of the **COMMUNICATOR** and its usefulness to our customers.

Please contact the editor if you have new ideas or can suggest changes in the content or format of the **COMMUNICATOR** which will foster the creation of a quality publication that meets the informational needs of HP Computer Systems users.

Address your correspondence to:

Editor Janis Andrews
Computer Systems Communicator
HP General Systems Division
5303 Stevens Creek Blvd.
Santa Clara, CA 95050

You may also contact the appropriate divisional technical editors as follows:

- HP 1000 System (9600/9700)
  Gary Gubitz
  HP Data Systems Division
  11000 Wolfe Road
  Cupertino, CA 95014

- HP 2000 Computer System
  Dan Jorgenson
  HP General Systems Division
  5303 Stevens Creek Blvd.
  Santa Clara, CA 95050

- HP 3000 Computer System
  Janis Andrews
  HP General Systems Division
  5303 Stevens Creek Blvd.
  Santa Clara, CA 95050

contents

software tips

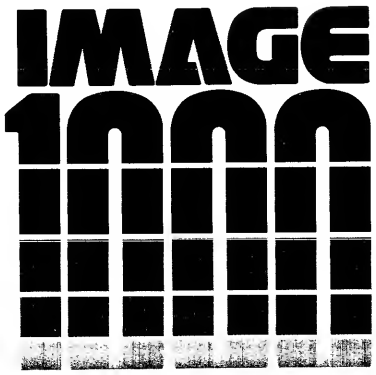


IMAGE 1000

Paul McGillicuddy
HP Data Systems

This article will suggest some programming hints for IMAGE. The basic assumption is that the IMAGE application will be segmented and therefore, the segments should contain the most optimum code.

STEP 1 First determining the size of your partition.

a. IMAGE will need room for the DCB of the File Manager. This will be a maximum of 1632 (which allows 6 Data Sets to be opened at one time with a buffer size of 256 words) or a minimum of 144 (which allows only 1 Data Set to be opened at a time with a buffer size of 128 words).

b. IMAGE also needs room for the root file. Its size can be computed as follows:

55 + (# of Data Sets * 16) + (# of items * 5)
+ (# of items + 2 * Path Count + 1

c. The space left is available for your program (main and segments). It is possible to determine how large a program can be by subtracting the above requirements from your partition size. Be sure to leave some area for future growth.

STEP 2 Determine the structure of the main program and its segments.

a. The main program should contain code that will be used most often. For example, a general prompting message to the user about his next function. It should not contain prompts to the user about the Data Base name, security code, etc., since this is done only once.

b. The segments should allow for some future expansion. With this in mind, determine the size of the largest segment. Remember that subroutine calls (direct or implied by a FORTRAN statement) are loaded as part of the main if they are called in both a main and a segment.

The following is an outline summarizing the above points:

MAIN PROGRAM

```
      ASSIGN 100 TO IRET
      CALL EXEC (8,ISEG)
  100 CONTINUE
          .
          .
C         .
C     DUMMY CALL TO LOAD DBBUF IN MAIN
C
  200 CALL AIDCB
      END
```

Using the ASSIGN allows a return to the main program from a segment. ISEG in this case will be INFOD. IRET must be COMMON.

AIDCB allows DBINT and DBOPN to be called in a segment.

SEGMENT 1

```
      PROGRAM INFOD(5,90)
C
C     PROGRAM TO GET INFORMATION ABOUT THE DATA BASE
C
      COMMON ICBUF(20)
      DIMENSION IBASE(3),ILEVL(3),ISEG(3)
      EQUIVALENCE (ICBUF(4),IBASE(1)),(ICBUF(7),ISCOD),
     1 (ICBUF(16),ILEVL(1)),(ICBUF(19),IMODE),
     2 (ICBUF( 1),LU)
      DATA ISEG/2HIN,2HOP,2HD /
      WRITE(LU,100)
  100 FORMAT("DATA BASE NAME = ?←")
      READ(LU,200)IBASE
  200 FORMAT(3A2)
      WRITE(LU,300)
  300 FORMAT("SECURITY CODE = ?←")
      READ(LU,*)ISCOD
      WRITE(LU,400)
  400 FORMAT("LEVEL WORD = ?←")
      READ(LU,200)ILEVL
      WRITE(LU,500)
  500 FORMAT("ACCESS MODE = ?←")
      READ(LU,*)IMODE
      CALL EXEC (8,ISEG)
      END
```

This segment prompts the user about the data base. COMMON is used to pass the answers to the next segment.

← suppress carriage return, linefeed

SEGMENT 2

```
      PROGRAM INOPD(5,90)
C
C     SEGMENT TO INITIALIZE AND OPEN THE DATA BASE
C
      COMMON ICBUF(20)
      DIMENSION IBASE(3),ILIST(4),ISTAT(4),ILEVL(3),ISEG(3)
      EQUIVALENCE (ICBUF(4),IBASE(1)),(ICBUF(7),ISCOD),
     1 (ICBUF(8),ILIST(1)),(ICBUF(12),ISTAT(1)),(ICBUF(16),
     2 ILEVL(1)),(ICBUF(19),IMODE),(ICBUF( 1),LU)
      DATA ISEG/2HCA,2HLC,2HQ /
      ILIST(1)= 1
      ILIST(2) = 2HCA
      ILIST(3) = 2HLC
      ILIST(4) = 2HQ
C
C     INITIALIZE THE DATA BASE
C
  100 CALL DBINT(IBASE,ISCOD,ILIST,ISTAT)
      IF (ISTAT.NE.0) CALL DBERR(1,ISTAT)
C
C     OPEN THE DATA BASE
C
  300 CALL DBOPN(IBASE,ILEVL,ISCOD,IMODE,ISTAT)
      IF (ISTAT.NE.0) CALL DBERR(2,ISTAT)
  500 CALL EXEC (8,ISEG)
      END
```

Because COMMON cannot be initialized at compile time, a DATA statement cannot be used to initialize ILIST. CALCQ is the largest segment in the program and is the only segment name required by DBINT.

DBERR is a status checking routine. It passes a number representing the current call to allow easier debugging.

An alternative to this method would be to have the Data Base Name, etc., and the largest segment name in a disc file that is read in by a segment to initialize the program's variables.

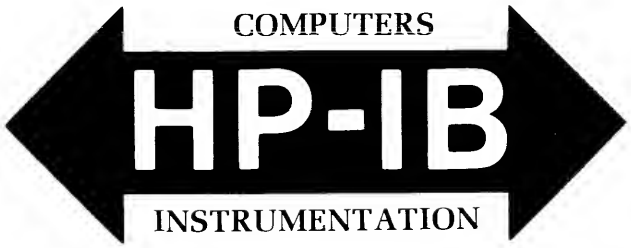
SEGMENT 3 to N

These segments perform the functions of the application. The largest should be 80% of the room left in the partition.

If any of these segments require a return to the main they can execute a : GO TO IRST, assuming they have a variable IRST in the corresponding position of the main.

SEGMENT N + 1

This last segment does the close.



HP-IB TREKIE

Performance Analysis Article #1

Now that the HP-IB scheme for interfacing digital equipment is known to the electronic world, the following series of articles titled

"HP-IB TREKIE"

will deal with common and useful things to do with an HP-IB system. The initial thrust and series of articles will deal with performance studies on different areas of HP-IB applications. The intent of these first six articles will be to analyze "WORST-CASE" conditions so the user will have a good idea of how far to push an HP-IB system. It is also intended to give realistic data on system configuration and operation. So, stay tuned to the **COMMUNICATOR**. There's a whole lot of interesting and surprising things to come in the HP-IB area. As one current HP-IB user commented, the term 'bus' really stands for

B BEFORE
U
S UFFERED

To begin our trip down the HP-IB path, let's summarize all the topics to be covered in this and future issues of the **COMMUNICATOR**:

THIS ISSUE	:	HP-CABLE PARAMETERS & DEVICE LOAD CONSIDERATIONS
NEXT ISSUE	:	BUS TOPOLOGY & HANDSHAKE ANALYSIS (DON'T MISS THIS ONE)
NEXT ISSUE +1	:	OPEN COLLECTOR DELAY CONSIDERATIONS
NEXT ISSUE +2	:	POWERED-OFF DEVICE CONSIDERATIONS
NEXT ISSUE +3	:	DATA SETTLING TIMES & DAV GLITCH
NEXT ISSUE +4	:	SUMMARY

CABLE PARAMETERS: A LOOK AT CABLE DISTANCE CONSIDERATIONS

The purpose of this article and ones to come is to present a selective and condensed version of a performance study conducted by Hewlett-Packard at Data Systems Division in August 1976 titled "HP-IB WORST-CASE ANALYSIS." It's purpose was to arrive at suggested system guidelines for optimizing HP-IB performance under worst case conditions. Since the HP-IB is essentially new from the applications end of Data Systems Division, it is hoped that these articles will prove to be useful for the HP-IB user in configuring and/or designing a system. It is also hoped that some clarification will arise from these discussions regarding any mysterious "HP-IB IF'S" concerning instrument clusters.

If you've had a difficult time interpreting the IEEE-488-1975 specification (especially if you're a software cat), then these articles might help eliminate any fear and/or preconceived misconceptions about the BUS. All in all, the terminology and illustrated diagrams have been kept simple and uncluttered with the normal rhetoric that seems to always accompany such discussions. I hope you will enjoy our articles as much as we have by putting them all together. If you have any questions regarding anything mentioned, please take time to write the editor so a quick reply can be initiated. We look forward to receiving your comments on our articles.

I would also like to take time to mention that any articles which you feel would be useful to other users will be taken with enthusiasm and considered for publication. Since you are our most valuable resource, we consider it a compliment when you write us.

The worst-case data used for the calculations and tests were derived from the HP-IB specifications as given in IEEE-488-1975, which will be hereafter referenced as "THE STANDARD".

Since the standard specifies only the maximum capacitance of a single signal line in the cable, the worst-case cable parameters are based on measurements of existing cable assemblies. Three measurement techniques were used in this study:

1. Direct measurement of inductance and capacitance using a 4260A RLC meter.
2. Propagation delay times over a long cable section using a dual-trace oscilloscope.
3. Characteristic impedance by resistor selection at the ends of a moderate-length cable and observing reflections on the oscilloscope.

The cable consisted of multiple 3-foot and 6-foot sections, HP parts 10631A and 10631B. The results were as follows:

MEASURED INDUCTANCE:	0.16 microhenry per foot
MEASURED CAPACITANCE:	40 picofarads per foot
MEASURED DELAY:	2.13 nanoseconds per foot
MEASURED IMPEDANCE:	65-75 ohms

The results of the three measurements agreed rather well as for an ideal transmission line, the delay and characteristic impedance are given by

$$T = \sqrt{LC} \qquad Z = \sqrt{\frac{L}{C}}$$

The measured inductance and capacitance thus predict the following values:

$$T = \sqrt{(0.16 \times 10^{-6} \times 40 \times 10^{-12})} = 2.53 \text{ nsec/foot}$$

$$Z_0 = \sqrt{(0.16 \times 10^{-6} / 40 \times 10^{-12})} = 63.2 \text{ ohms}$$

Based on this correlation and the relative confidence placed in the measurement techniques, the typical values are assumed to be

L = 0.147 microhenry per foot = 0.482 ohms per meter
C = 40 picofarads per foot = 131 pf per meter
T = 2.42 nanoseconds per foot = 7.95 ns per meter

You might note that the standard allows 150 PF per meter of cable as the worst-case capacitance, but the worst-case inductance is not specified. Using the ratio of typical capacitance to worst-case capacitance as a rough indicator, the worst-case inductance was assumed to be 0.6 microhenry per meter and the resulting worst-case delay becomes 9.5 NSEC per meter. Watch out for this! Since the HP-IB is an industry standard, any company peddling IB-compatible products does so at the risk of answering to any design deficiencies in the product and/or deviations from the standard.

DEVICE LOAD CONSIDERATIONS

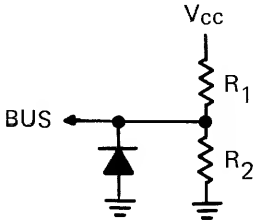
There are really two things to consider: capacitance and resistive termination. In the area of capacitance, the standard permits a device capacitance of 100 PF on each signal line. Since this capacitance is expected to have a significant effect on the rise times of the open-collector handshake lines (RFD and DAC) and also the cable delay derived above, measurements were taken on a few existing HP-IB devices to determine whether 100 PF is a reasonable value. The measurement apparatus consisted of an open-collector driver with resistive pull-up, allowing the R-C time constant to be computed from the exponential waveform on the signal line in question. Results with a 9871A Printer, 59303 D/A Converter, and 3495A Scanner indicate that 100 PF is an unreasonable high value since all three devices responded with capacitances well within 50 PF. As a result, 50 PF is assumed to be the worst-case device capacitance.

The standard resistive termination requires each device to terminate each signal line with a resistive load as is specified by a region on a voltage-current graph in IEEE-4888 specification. Since the equivalent resistance will be greater than about a thousand OHMS, its main effect on system performance will be to help determine the rise time of the open-collector signal lines. Therefore, the worst-case resistance is one which provides the minimum current to charge up the cable and device capacitances. Since the standard permits this minimum current to be arbitrarily small in the region between 0.4 volts and 2.5 volts, this means that the open collector rise times could be arbitrarily long. While such a situation probably will not occur, the standard would make some mention of it.

For the purposes of this study, the worst-case resistance is assumed to be defined by the straight line between the two points

0.4V, 1.3 ma
and 2.5V, 0 ma

The equivalent circuit would be



It might be noted that this circuit would actually occur if a high input impedance receiver circuit were used. The resistor values may be found by solving for the current onto the signal line with

$V_{cc} = 4.75$ volts

Thus,

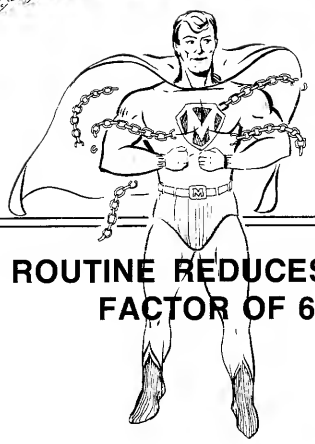
at 0.4 volts: $1.3 = \frac{4.75 - 0.4}{R1} - \frac{0.4}{R2}$

at 2.5 volts: $0 = \frac{4.75 - 2.5}{R1} - \frac{2.5}{R2}$

The resulting worst-case values are R1 = 3.07K and R2 = 3.41K.

In the next issue, typical device configurations will be discussed along with speed guidelines and specific device time requirements.

MICROPROGRAMMING



MICROPROGRAMMED SORT ROUTINE REDUCES PROGRAM EXECUTION TIME BY FACTOR OF 6

Gary Gubitz
HP Data Systems

:RU,ASORT,1,5000
START OF SORT
END OF SORT

ASSEMBLY LANGUAGE SORT WITH 5000 NUMBERS

	Hours	Minutes	Seconds	
STOP:	8	29	11.26	19 sec. on 21MX M
START:	8	28	52.35	11 sec. on 21MX E

:RU,MSORT,1,5000
START OF SORT
END OF SORT

MICROPROGRAMMED SORT WITH 5000 NUMBERS

	Hours	Minutes	Seconds	
STOP:	8	29	38.95	2.2 sec. on 21MX M
START:	8	29	36.69	1.75 sec. on 21MX E

Are you looking for a better way to accomplish your applications program tasks? Have you used all the programming methods you can think of to make your library subroutines run as efficiently as possible in your Real Time Executive (RTE) operating system environment? Well then, maybe it's time to look into Microprogramming.

What is Microprogramming? Primarily, Microprogramming is the modification of and/or addition to the computer instruction set without hardware redesign so that you may have the advantage of executing selected main memory programs at the fastest possible rate available in the computer.

In other words, Microprogramming allows you to increase the horsepower available from your computer without having to upgrade processing units.

Why Microprogramming? Because Microprograms and Microprogramming techniques can be used to:

1. Reduce program execution time:
   - Many instruction fetches are eliminated.
   - Multiple operations can occur during a single microinstruction.
   - The microinstruction word width (24 bits) provides a larger instruction repertoire than available with the Assembler word width (16 bits).
   - Many more registers and functions at the microinstruction level are available to you than to the assembly level or high level language programmer.
2. Implement customized computer instructions:
   - Special arithmetic instructions (double integer, decimal, etc).
   - Emulation.

WHAT TYPES OF APPLICATIONS CAN BE MICROPROGRAMMED?

- Sort Routines (e.g., BUBBLE, SHELL, RADIX-EXCHANGE, and QUICKSORT).
- High-Speed or specialized input/output transfer operations.
- Table searches (e.g., sequential, binary, and link-list).
- Transcendental functions (e.g., sine, square root, and logs).
- Fast fourier transforms (FFT).
- and many, many more!

Once it is determined that Microprogramming can make your operation a more efficient one, the question then becomes, "What should be microprogrammed?" What is

needed is some method of determining what to microprogram. Let us take a look at one such method.

In the Hewlett-Packard Contributed Library, there exists a program that performs an activity profile generation under RTE on any program(s) you desire to analyze. The idea behind this is to determine where your application program is spending most of its time, and then, once isolated, these areas become the candidates for Microprogramming. A sample output from an activity profile generation follows: [Note: We monitored a program called TRIG].

Activity profile generation for TRIG

Memory Analyzed From 50000B To 51374B.
51 Memory Cells, each 00017B long;
0 activity cells not listed.

LOCATIONS	INTERRUPTS	% USAGE	CUMULATIVE % USAGE
UNDER	1.	.02	.02
50000-50016	124.	2.54	2.56
50036-50054	171.	3.50	6.06
50606-50624	213.	4.36	10.42
50625-50643	145.	2.97	13.39
50644-50662	306.	6.27	19.66
50663-50701	84.	1.72	21.38
50702-50720	4.	.08	21.46
50757-50775	261.	5.34	26.80
50776-51014	258.	5.28	32.08
51015-50133	504.	10.32	42.40
50134-51052	429.	8.78	51.19
51053-51071	204.	4.18	55.36
51072-51110	56.	1.15	56.51
51130-51146	98.	2.01	58.52
51147-51165	65.	1.33	59.85
51166-51204	100.	2.05	61.90
51205-51223	796.	16.30	78.19
51224-51242	812.	16.63	94.82
51262-51300	61.	1.25	96.07
51301-51317	46.	.94	97.01
51320-51336	105.	2.15	99.16
51337-51355	16.	.33	99.49
51356-51374	25.	.51	100.00
OVER		.00	100.00

Total Samples = 4884.
Execution Time = 48.84 seconds
Sampling Frequency = 100. per second

Program	Relative CPU % Utilization	Samples
TRIG	100.0	4884.

By taking the output from this activity profile generation, and comparing the % usage column with the load map of the program being monitored (TRIG in this case), one can easily determine the candidates for Microprogramming (the boxed items).

The capabilities and potential application areas for Microprogramming exist, what is needed is a tool to enable *YOU*, the user, easy access to these capabilities.

The RTE Microprogramming Package is that Tool!

The RTE Microprogramming package allows the user to take full advantage, with relative ease, of the Microprogramming capabilities of our Hewlett-Packard 21MX M-Series and 21MX E-Series computers, and our most widely used operating systems, RTE II/III.

The features of the Microprogramming package are:

- On-line operation in RTE-II/III systems.
- Cross reference generator for simplified program development.
- Microdebug editor for interactive program editing and checkout.
- Operator entered breakpoints.
- Full writable control store support, including driver, load utilities, and load verification routines.
- Dynamic writable control store overlay utilities.
- Software support for both 256 and 1024 word writable control store.
- Prom tape generator for outputting production microcode on (punched) prom "BURN" tapes in formats most commonly used by vendors.

Remember ** Hewlett-Packard ** has a complete Microprogramming package!

- HARDWARE — Writable Control Store; Ability to Utilize Prom Chips
- SOFTWARE — Microdebug Editor; Microassembler; WLOAD (WCS Utility); DVR36 (WCS Driver); PTGEN (PROM Mask Tape Generator); MXREF (Cross-Reference Generator)
- TRAINING — 4 Day Applications Course
- SUPPORT — Local and Central

INTERESTED? For more information concerning the RTE Microprogramming package contact your local Hewlett-Packard representative.

In future issues we will continue to write informative material about Microprogramming. After all, we are here to help you solve problems. If you have any suggestions for topics, let us know.

WRITE TO: MICROPROGRAMMING
c/o Communicator H-P 1000 Group
Hewlett-Packard Data Systems Division
11000 Wolfe Road
Cupertino, Ca. 95014

HINTS ON EFFECTIVE USE OF RTE SOFTWARE

Jim Bridges
HP Data Systems

Experienced users have discovered several techniques which not only make more effective use of disc space but also tend to make changes to a system self-documenting. The techniques have evolved over a period of years and have been passed along verbally. It is believed that this is the first time they have appeared in any publication distributed outside of Hewlett-Packard.

The term "RTE" will generally mean RTE II (92001A) or RTE III (92060A) but many of the concepts expressed are applicable to the original RTE (2300E) as well. Extensive use is made of the file management package (FMP) and so they do not apply to very minimal systems. The assumption is made that the system may have several programs added after generation and that the system supports several users. The system may also have several ports operating under the multi-terminal monitor (MTM).

The techniques will be presented serially, in the same order that they would be used in the life of a system.

SYSTEM GENERATION

1. Add a program called SYSON to the system at the time of generation. This program is designed to be the "start-up" program, i.e., scheduled automatically at boot-up. Usually this program is referenced during the parameter entry phase by adding 80 to the type. For example:

   SYSON,82,50

   which makes SYSON a foreground disc resident program with priority 50. The type (2) is added to 80 to give 82.

   The many different SYSON programs all have one feature in common: at one point they schedule FMGR and pass to it the file name WELCOM. A typical WELCOM file will be presented later. A sample SYSON program is:

```
      PROGRAM SYSON
      DIMENSION NAME (3)
      DATA NAME/2HFM,2HGR,2H /
C SCHEDULE FMGR WITHOUT WAIT IN QUEUE:
      GO TO WELCOM FILE
      CALL EXEC (24,NAME,2HWE,2HLC,
      2HOM,0,0)
      END
      END$
```

2. Obtain the system generation map in machine readable form. With the off-line generator, this may be accomplished by using a CRT as the console and the list device and echoing the questions and answers on a high speed punch. Later, this tape will be stored in a file called "SYSTM for reference.

FIRST BOOT-UP

When the system is booted the first time, the message

FMGR 002

requests initialization of the FMP area of logical unit #2 (LU 2). If LU 3 is not zero, a similar message will be given for LU 3. After LU 2 and 3 are initialized, the message

FMGR -006

indicates, in this case, that the WELCOM file has not been found. The name of this file was passed to FMGR by SYSON. Now is the time to create the WELCOM file. The sample WELCOM file assumes that there are two MTM terminals (LU 7 and LU 9).

```
:SV,1                    (Inhibit command echo)
:DU,HELLO:RT:2,1         (Dump a message indicating
:DU,HELLO:RT:2,7           system start-up)
:DU,HELLO:RT:2,9
::*COPY,FMGR,FM07.       (Make copies of FMGR for LU 7
::*COPY,FMGR,FM09.         and 9)
::*COPY,EDITR,ED07.      (Make copies of EDITR for LU 7
::*COPY,EDITR,ED09.        and 9)
:PU,FILL02:RT:2          (Purge dummy files to control
:PU,FILL03:RT:3            available space on LU 2 and 3)
::*PACK                  (Pack files on LU 2 and 3)
:CR,FILL02:RT:2:4:-1     (Create dummy files to protect
:CR,FILL03:RT:3:4:-1       space on LU 2 and 3)
:DU,READY:RT:2,1
:DU,READY:RT:2,7         (Tell users that we're ready
:CN,7,20B                  and enable their terminals)
:DU,READY:RT:2,9
:CN,9,20B
:SV,0
:EX
```

Many points are illustrated in this sample WELCOM file:

1. Reference is made to auxiliary files HELLO, READY, *COPY and *PACK. HELLO and READY message files are tailored to whims of the composer. *COPY and *PACK are more specific:

```
:RN,1G:RT:2,2G           (This is the *COPY file)
:RP,2G                   (An ID segment is given to
:RN,2G:RT:2,1G             the new file and the origi-
::                         nal file name is restored)

:PK,2                    (This is the *PACK file)
:PK,3
::
```

One might wonder why the pack commands are put in a separate file. The reason is that the WELCOM file itself might be moved during the packing. A WELCOM file could cross sector boundaries after the appearance of the pack commands. In this case, succeeding commands might be read from the wrong area of the disc. But, by taking the pack commands from a small file (*PACK) which fits in a single sector and then returning to the WELCOM file, the WELCOM file is accessed from its new location (if moved).

2. The WELCOM file expects FMGR and EDITR to be saved in type 6 files of the same name. A copy of each is made for each MTM terminal by the convention that the first two characters of the name are retained, the second two characters match the LU of terminal and the last two characters are ". ".

3. LU 2 and 3 are initialized so that their cartridge numbers are 2 and 3 respectively.

4. Files with universal meaning (such as FMGR, EDITR, WELCOM) are protected from inadvertent destruction by a "universal security code" of RT (first two characters of RTE).

5. Dummy "fill" files are used to control the use of space on LU 2 and 3. This is done because a type 6 file can be given an ID segment only if it is on LU 2 or 3. Also, because space on LU 2 and 3 is recorded in the track assignment table (TAT), it is not truly "general purpose". Any file which could reside on another disc subchannel should reside on another disc subchannel.

When the WELCOM file has been created, the next step is to store the generation map in a file called "SYSTM. For example:

:ST,5,"SYSTM:RT:2::-1

Since the map is usually a large file, it may be desirable to put it on an LU other than 2 or 3, especially a peripheral disc subchannel that can be backed up on magnetic tape.

Mount and initialize the peripheral disc subchannels. If the system has enough disc space and there are several MTM terminals, it is a good idea to "logically associate" each terminal with a specific subchannel. For example, put an identifying tag on each terminal which tells the user of that terminal which disc LU he should use. There is nothing to actually prevent him from using another LU, but if each user agrees to the rules, interference between users can be minimized.

Re-boot the system to test out the WELCOM file. If all is well, then individual users can begin work. However there are some other conventions they should agree to use in adding programs.

FILE NAMING CONVENTIONS

If all agree to use a convention for naming files, it is much easier to examine the contents of a disc. This will be true even if you are the creator of the files because, as time goes on your memory dims. A convention in wide use is to attach significance to the first character of a file name as follows:

First char	Meaning
&	Source code for a program
%	Binary relocatable object code
!	Binary absolute object code
/	Set-up file with FMGR commands
\	Clean-up file with FMGR commands
"	Documentation or information file
*	General purpose transfer file with FMGR commands

A file name which begins with a letter is either a type 6, the WELCOM file (or a file used by WELCOM) or a temporary file. Temporary files should not appear on LU 2 or 3. The WELCOM file (and any files it uses) and type 6 files should always be on LU 2 or 3 and should have security codes.

It is useful to be able to tag different revisions of the same program. There are two good ways to do this:

1. Identify source and relocatable binary which is in the development stage by alternate first characters in the name. Typically, $ is used as an alternate for source and # is used as an alternate for relocatable binary. The use of & and % would indicate the "debugged" software.

2. Identify the file with a date code by making use of the record length field in the NAMR. The record length is required only for type 2 files. It is optional for type 3 and above. It is recorded in the directory entry and listed when the command to list the directory entry (:LI,< namr >,D) is given. The record length is limited to 4 decimal digits. For example,

   :CR,&SOURC::15:4:48:100S

   creates a source file (type 4) of 48 blocks on CR 15 with a record length of 1005. FMP will not use the record length (type 4 files have variable record length) but I can choose that it mean the month and day. In this case, that would be 10/05 (October 5).

The EDITR program does not pass the record length to FMP so source files created by EDITR will have to be stored in the file to make use of this feature. For example:

```
:ST,&SRC1::15,&SRC2::15::-1:1005
:PU,&SRC1::15
:RN,&SRC2::15,&SRC1
```

ADDING PROGRAMS TO THE SYSTEMS

There are two ways to add programs to the system. One is to make them permanent additions. The other is to make them temporary and to save them in type 6 files. There are four major reasons why the latter approach is recommended:

1. It conserves ID segments. The program may be purged after the type 6 file is established. This releases the ID segment for use. Many type 6 files may, in sequence, reuse the same ID segments.

2. It conserves disc space. Programs added by the on-line LOADR program always begin on sector zero of a previously unused track. Each segment will begin on the first even-numbered sector following the main or the last-loaded segment. However, type 6 files begin at the next unused even-numbered sector in the FMP area. Type 6 files are also packed (the same as other file types) by the :PK command.

3. It keeps the amount of scratch area constant. Initially, the scratch area is the space between the last track reported used by the generator and the last track on LU 3 – excluding the FMP area on both LU 2 and 3. Space for programs added on-line are taken from the scratch area – diminishing this valuable resource. Scratch tracks are required for program swapping, LS and LG areas and are temporarily assigned to executing programs requesting disc tracks.

4. Programs which are saved in type 6 files may be renamed and used from several different terminals. The overhead for each terminal is an ID segment in memory and swap tracks for each active copy.

To assist in adding programs on-line (saving them in type 6 files), a general purpose transfer file called *SP is created as follows:

```
:PU,FILL02:RT:2
:SP,1G:RT:2
:OF,1G
:CR,FILL02:RT:2:4:-1
::
```

For example, let's assume the FORTRAN IV compiler (24177-60001) is added on-line rather than generated into the system. The compiler has four segments. After the LOADR has loaded FORTRAN as a temporary program, do the following:

```
::*SP,FTN4      (Build type 6 files for FORTRAN IV and
::*SP,F4.0        its four segments)
::*SP,F4.1
::*SP,F4.2
::*SP,F4.3
```

Then create a set-up file and a clean-up file.

The set-up file, following the name convention is, /FTN4

```
:RP,FTN4        (Build ID segments for FORTRAN IV)
:RP,F4.0
:RP,F4.1
:RP,F4.2
:RP,F4.3
::
```

The clean-up file, following the name convention, is \FTN4

```
:RP,,FTN4       (Release ID segments for future use)
:RP,,F4.0
:RP,,F4.1
:RP,,F4.2
:RP,,F4.3
::
```

This concludes the discussion of helpful techniques in using and maintaining an RTE system. Please let me know if you have any suggestions to add. I will publish them in follow-up articles with appropriate thanks to contributors.

CAUTION ON DEFINING 7905 SUBCHANNELS FOR RTE-II/III

Jim Bridges
HP Data Systems

The generators for RTE II (92001A/92001B) and RTE III (92060A/92060B) do not check for certain user errors in defining 7905 subchannels. These errors can result in one or more tracks "shared" by different subchannels. The consequences can be catastrophic. The following example makes the point:

# TRKS	FIRST CYL #	HEAD	# SURFACES	UNIT	# SPARES
255	0	2	1	0	4
150	259	2	1	0	2
400	0	0	2	0	11
400	205	1	2	0	11

Subchannel 3 starts on head 1 and uses *two* surfaces. The generator *increments* the head number to determine the heads to use for the two surfaces. Thus, subchannel 3 uses heads 1 and 2. But this means that subchannel 3 and sub-

channel 1 have tracks 205 through 259 and head 2 in common! The generator assumes that any application of the disc is valid and so this is not flagged as an error. But if subchannels 1 and 3 are used for FMP files (or the system and FMP files) files may be lost or the system destroyed by itself.

The next update to the system manuals will clarify the method used by the generator so that future readers may avoid this trap.

KNOW YOUR RTE — PART 5

by Mr. RTE

This is the fifth of a series of articles in the **Communicator** dealing with the inner works of HP's RTE systems. These articles go into some detail on how the system works; therefore, you should have already read and become familiar with the material in the RTE reference manual to your system.

Last time we finished up tracing a command through $MESS. We now have enough of the system within our grasp to attempt an I/O request. An I/O request is made by a program doing a JSB EXEC. EXEC is below the memory protect fence so the JSB to EXEC will cause a memory protect violation. This causes an interrupt to select code 5. Trap cell 5 contains a JSB to $CIC in RTIOC. $CIC saves the calling programs registers, identifies the interrupting select code and if not 5 clears its flag. $CIC then sets up the privileged interrupt system if the privileged option was chosen at generation.

At this point $CIC further catalogs the interrupting select code as follows:

1. If it is 5 go to $RQST in EXEC
2. If it is the time base generator, go to $CLCK in RTIME
3. If in the interrupt table then either:
   a. If entry is > 0, it is an EQT address – set it up and go to the driver at C.XX
   b. If entry is < 0 then it is the negative of the ID segment address of a program to schedule.
   c. If entry is 0 print ILL INT XX and return to the point of interrupt.
4. If none of the above, print ILL INT XX and return to the point of interrupt

In the case we are tracing (an I/O call), the system goes to $RQST in EXEC. $RQST reads the violation register, sets it as the current point of suspension, checks for a dynamic memory error (RTE-III) and for parity error (Halt 5, B= violation address). If these tests fail, then the violating instruction is tested for the following:

1. It must be a JSB or JSB,I.
2. The target address must be EXEC, $LIBR or $LIBX or the target address must be in the resident library.

If either of these fail, a MP error is reported and the program is aborted by calling $ERMG. We will digress at this point to discuss $ERMG.

$ERMG is entered with the 4 character ASCII error message in the A and B registers. For example A might = IO and B = 01 to report error IO01 XXXXX AAAAA where XXXXX is the current executing programs name (from the ID segment pointed to by XEQT on the base page) and AAAAA is that programs current point of suspension (XSUSP,I on the base page). After the message is set up, but before it is sent, a check is made to see if the B-register portion of the error code is a double blank. If it is not a double blank, the programs status word is checked to see if Bit 15 (the NA bit) is set. If this bit is set, it means the program has made a No-Abort EXEC call. (We will see how the bit gets set below). If the NA bit is set and B ≠ double blank, then the programs A and B registers are set to the error code (A and B at entry to $ERMS), its point of suspension is decremented and the program is scheduled by calling $LIST. $ERMS then returns.

In the case where B=double blank (e.g., MP, RQ, DM) or the NA bit is zero, the error message is sent to the system console via $SYMG in RTIOC. The message indicates why, who and where. The $ERMS routine then calls $ABRT (see part 4) which generates the XXXXX ABORTED message. $ERMS then returns to its caller.

If we have a good JSB EXEC the system uses the return address from the EXEC call to calculate the number of parameters in the call. It will abort the program with an RQ error if:

1. There is less than 1 or more than 9 parameters
2. If a parameter address links through either A or B

If these tests pass a check is made to see if the sign bit is set on the request code. If it is the NA bit is set in the programs status word, the return address is incremented and the sign bit is cleared in the request code. Most EXEC request processors don't even know about the abort return option as this code and the code in $ERMS do all the work. This request code is put in RQP1 on the base page so that we can modify it without changing the users copy. All other parameters are left as DEF's on the base page. Any unsupplied parameters have their DEF's set to zero on the base page.

Next the system checks for a legal request code and aborts the program with an RQ error if it is illegal.

The system then indexes into a parameter bit map which identifies which parameter, for each request code, the system may store into. Each of these parameters is checked to see if the supplied address is above the current memory protect fence. If not the program is aborted with an RQ00 error. If the parameter is to be altered but is not supplied, the system ignores the error at this point. If it is truly an error, the processor for that request will report it.

After these checks the I/O request we started with is dispatched to $IORQ in RTIOC.

$IORQ first picks up the LU, then checks the batch flag (BA in word 21 of the callers ID-segment). If the batch flag is set, the LU switch table is scanned to find the current LU in the 'from' side of the table. If found the LU is translated. After a possible translation, the LU is checked to see if it exists (i.e., is less than LUMAX on the base page). By holding this check until after the transform we allow a batch program to use an LU greater than LUMAX. The batch flag is set by FMGR in its own ID-segment and is propagated by the system whenever it schedules a program with wait so even the lowest son will have the batch bit set if it is part of a JOB.

If the LU equals 0 or if it points to EQT 0, RTIOC sets up a dummy EQT in RTIOC itself. This dummy EQT has a select code of zero so a two-word status request can identify it. In addition to the status words EQT1 and EQT6 are needed by the system, thus the system uses 4 words to put together a 15 word dummy EQT.

If the EQT is real $CVEQ is called to set it up. At this point the status request (request code=13) is split out. We will not cover the status request in any more detail in this series.

At this point a check is made to see if the EQT or the LU is down. If it is, the first temporary word in the callers ID segment is set to 4 and the program is put in the 3 list ($LIST) and RTIOC exists to $XEQ.

When any device is upped, the programs in the 3 list with their first temporary word set to 4 will be rescheduled, but for now, the program is suspended and swappable.

Control requests; i.e., request codes 3 or 19 (by this point you should have observed that class request codes are just 20 octal more than standard I/O requests – not by accident) are handled next. The control requests are fairly straight forward and we won't do more than say the dynamic status request is never buffered. This allows you to get the status after all the other stuff queued on the device has been done and to get it even if the device is buffered.

For other I/O requests the system checks for at least 3 parameters and for standard READ and GET requests that the buffer does not extend beyond the end of memory (IO04 if it does). If the device is a disc (driver type code= 30,31,32 or 33) and the request is a CLASS request, the program is aborted (IO02). If the LU is 2 or 3 (i.e., system discs) there are 2 or 3 pages of assembly language tests that are done to make sure the access is legal. We will not detail these here. For non-disc I/O, if the user has set bit 12 (X-bit) he is supplying two buffers (see your RTE manual). The system assumes that this buffer will not be modified so no write protect checks are done on it but if the bit says it is supplied then there must be 5 parameters (IO01). At this point the LU lock flag is checked. If the LU is locked to some other program the current program is put into the 3 list ($LIST) to wait for the RN (resource number) associated with the LU lock. If the LU is locked to the current caller the system sets up to link the request as if it had the lowest possible priority (77777B) and to set the priority of the request itself to 1 (i.e., as high as possible). The reason for this is that if you're doing a block of I/O requests to a locked device, you want the requests to come out in the order they are made, but if there are requests from some prior user already queued on the device when the lock was granted you don't want to get ahead of his requests. This is insured by linking at the lowest possible priority. On the other hand you don't want some other user to get ahead of your final requests just because you unlocked the device before all the I/O in its buffer was completed. This is insured by giving the request itself the highest priority.

After some class I/O checks the system checks if the buffer limit has been exceeded. If the requesters priority is lower than 40 or the device is not buffered this check is not made. The device queue is checked and the number of words of buffer memory in it is summed. If the limit is exceeded the caller is suspended with his XTEMP word pointing to the EQT. When the device works the buffer down to the lower limit he will be rescheduled and will remake the I/O request.

At this point we rejoin the disc I/O, which is not checked for LU lock conditions. If the device is buffered and the request is for write or control, or if a class I/O request, then memory allocation is attempted. If there is no memory available now the program will be memory suspended to wait for memory. If there will never be any memory the system attempts to queue the request as unbuffered – see below. If memory is available the I/O request is set up in the buffer obtained – including a buffer move and if the X-bit is set the addition buffer is also moved. The request is then linked (using LINK an internal subroutine) on the EQT and control is given to $XEQ which will return to the user, or if no I/O is active on the device the internal subroutine DRIVR is called and then control is given to $XEQ.

For unbuffered requests (including those that could not be buffered as discussed above), the system checks the RE bit in word 21 of the callers ID-segment. If this bit is set the

caller is one or more levels deep in a re-entrant subroutine. If this is the case, the I/O buffer may be in that subroutines Temporary Data Block (TDB). If this is true and if, while the I/O is in progress, the subroutine is reentered the system would have to move a TDB while it was in use for I/O. In order to avoid this possibility, RTIOC calls $REIO in the EXEC module. This subroutine is passed the buffer address and checks to see if it lies within a TDB owned by the current program. If it does an attempt at memory allocation is made ($ALC). If there is no memory now but there will be at some time later (remember $ALC knows this), the program is memory suspended and $REIO exits to $XEQ instead of returning. If there will never be enough memory $REIO just returns to RTIOC. If memory is available the TDB containing the buffer is moved to system memory. This information must be passed back to RTIOC so that it knows where the buffer is.

This is complicated in RTE-III by the fact that it is possible that the actual buffer address has not changed even though the buffer has been moved to the system map. In the RTE-III this information must be known to correctly map the system when the I/O device interrupts so the entry point $MVBF is set non-zero to indicate that the buffer was moved. RTIOC also calls $REIO to move the second buffer if the X-bit is set. It is important to realize that both buffers must be in the same map or it will be impossible to correctly map the system for the driver. Also swapping is done based on the primary buffer address only — so if a second buffer is supplied it must be in the same status as the first buffer. The system can handle these buffers in two different TDB's but not one in and one out.

We will defer the details of reentrant subprogram management until a later article.

After $REIO returns RTIOC finishes set up of the users ID segment, calls $LIST to I/O suspend the caller, and calls the internal subroutine LINK to link the request on the EQT. If the EQT shows the device is busy, control is passed to $XEQ at this point.

DRIVR

If the device is not busy RTIOC calls DRIVR to start the I/O. DRIVR is an internal subroutine in RTIOC. DRIVR allocates a DCPC channel, if required, sets up the device EQT with the request data and calls the device driver to start the I/O. After the device driver returns to DRIVR it returns. If DCPC channel is required but not available the DCPC wait count is incremented and the EQT is put in DCPC wait status. DRIVR also has a not ready return which causes RTIOC to print the not ready message and to requeue all requests as follows:

1. Standard I/O. The caller is put in the general wait list ($LIST) until the device is upped.
2. Class, buffered and system I/O. The requests are queued on the LU which is put down.

Dummy EQ or LU

If a request is made at LU 0 or on an LU which is unassigned (points to EQT # 0) the system treats the request as a dummy request by a) not calling DRIVR and b) by returning a completion indication with as many words transferred as requested.

This concludes our discussion of the I/O request. Next time we intend to cover a time base tick. We have covered most of the core resident system so far so we are looking for more topics. If you have any suggestions please send them to us.

Letters for Mr. RTE:

It seems we have some very scrutinizing readers:

Dear Mr. RTE,

In your Slow Boot article in Issue #8, May, 1976, of the Communicator, on page 377 Column E, I have found what I think may be an error. For DVR 31 — paper tape — , the boot-extension JMP-sys-loc should read X7505 rather than X7605.

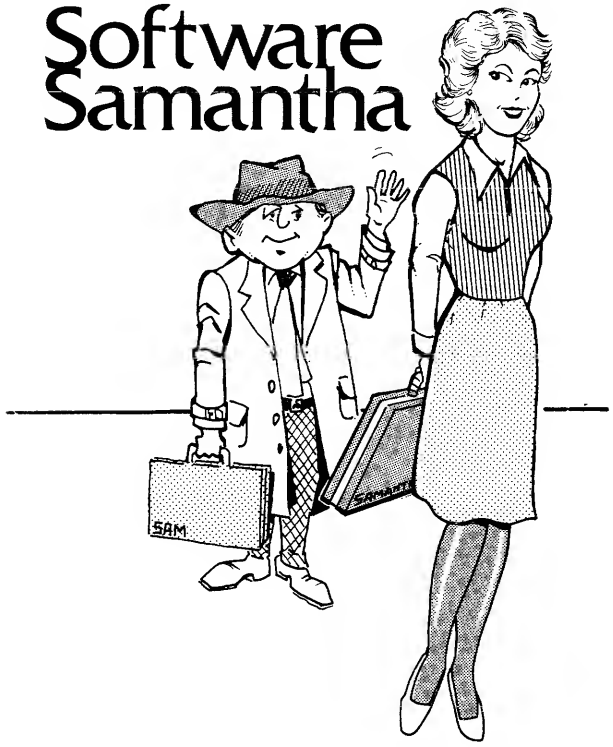
Del Kittendorf
HP Orlando

Dear Del,

You're right! Thank you.

Mr. RTE

Software Samantha



We would like to welcome to the **Communicator** the new 1000 (9600) system troubleshooter, Software Samantha, who is ready and willing to tackle problems or questions you may have with your 1000/9600 system. Samantha will respond to all letters sent in, the best of those being chosen for print. Also from time to time, she will share with you her insights in how to get the most out of your system.

The following is the first part of a letter addressed to Samantha's predecessor, Sam. The sender suggested a possible solution to the question below, that of saving the parameters in common area which is not swapped out with the overlay. Samantha will present a solution to the question of how to use any set of subroutines in an overlay segment and not lose parameters in the swap.

DEAR SAM,

If the plotter calls are to be added to Multi-User Real Time BASIC, RTE must be generated with at least 7503 octal words of foreground disc resident area to accommodate the 92409-60001 Rev C 7210A Plotter Library. This large foreground reduces the amount of background available for BASIC and BASIC programs. The Multi-User Manual (92060-90016 CHG 1) explains on page 14-6 how to load the overlay in background but it doesn't work because the PLTLU(LU) is wiped out as soon as BASIC is swapped back into background and before the plotter is commanded to move.

Bill Otto
DRSMI-RRS
Army Missile Command
Huntsville, Alabama

DEAR BILL,

I would first like to state that the problem is cleared up in revision D of 92409-60001. However, I will take this opportunity to explain the reason the problem occurred and how it may be avoided in revision C. I will address the general question of how one can use any arbitrary set of subroutines in an overlay segment which is swapped in and out of memory. This becomes a problem only if parameters are initialized by one subroutine call and must later be used by other subroutines in the overlay.

The following example will demonstrate the problem and the solution. Below is the assembly language code for two subroutines to be placed in a background overlay which will be assumed to swap in and out of memory, perhaps exchanging places with BASIC. The routine SETUP places a value in a location within the overlay. Later, the contents of the location are retrieved by calling LOOK.

```
ASMB,R,L
        NAM  SETUP
        ENT  SETUP,LOOK
        EXT  .ENTR
PARM    BSS  1
SETUP   NOP
        JSB  .ENTR
        DEF  PARM
        LDA  PARM,I
        STA  FLAG
        JMP  SETUP,I
FLAG    DEC  0
P1      BSS  1
LOOK    NOP
        JSB  .ENTR
        DEF  P1
        LDA  FLAG
        STA  P1,I
        JMP  LOOK,I
        END
        END$
```

The problem which BASIC must handle is this: if the overlay is swapped out between the time SETUP is called and the time LOOK is called, the value stored by SETUP will be lost if a fresh copy of the overlay is brought into memory. This may be avoided if the overlay terminates saving resources. When the overlay is accessed again, it is merely rescheduled instead of reloaded. Execution will then begin at the code immediately following the exit sequence instead of the primary entry point.

When a set of routines are written to be BASIC callable they normally include a call to another routine called RSFLG if they are saving parameters for later use. The

routine RSFLG is found in the relocatable library 24998-16001 (any rev), or 24248-60001 rev C. RSFLG sets a flag which is examined by BASIC to determine the method of exit. If RSFLG has been called within the overlay, the flag is set and the termination is made saving resources.

If RSFLG is not called by any subroutines in the overlay, it is still possible to make the overlay terminate saving resources. This is done by defining the routine RSFLG in the branch and mnemonic table itself. RSFLG must be called from the BASIC program before any other calls are made to subroutines in that overlay. For example, consider the code below:

```
10 CALL   RSFLG
20 CALL   SETUP(10)
30 CALL   LOOK(X)
40 PRINT  "X="; X
50 END
```

The branch and mnemonic table could be set up as follows:

```
BTBL, MTBL, TRFL, ID=A

SETUP(I),   OV=0,  ENT=SETUP
LOOK(IV),   OV=0,  ENT=LOOK
RSFLG,      OV=0,  ENT=RSFLG
```

The transfer file, TRFL, will be created by default to produce an overlay in the foreground area. It must be edited so that the overlay is generated as a background program. If the system is RTE II this will be sufficient to guarantee swapping between BASIC and the overlay. If it is RTE III, both BASIC and the overlay may have to be assigned to the same partition to demonstrate the problem. To see the difference between saving resources and "normal" termination, remove line 10 in the sample BASIC program. The result of line 40 will be "X= 0" instead of "X= 10".

If you have any questions, suggestions, or comments about your 1000 (9600) system, please address them to:

Software Samantha
c/o Communicator 1000 (9600) Group
HP Data Systems Division
11000 Wolfe Road
Cupertino, CA. 95014

bulletins

HP 9600/9700 SYSTEMS AND SOFTWARE UPGRADES TO HP 1000

Dave Borton
HP Data Systems

The introduction of the HP 1000 Computer Systems provides a new way to get RTE based systems. This new way includes the existing product capabilities of RTE-II and RTE-III as well as measurement and control. Measurement and control is now available as a computer system (HP 1000 for disc based systems and HP 9640A for memory based systems) plus a measurement and control front-end. For instance, a 9603A is now orderable as an HP 1000 plus a 9603R measurement and control subsystem with an option for local operation. Therefore, all 9600 and 9700 systems except the 9640A will be removed from HP's price list on November 1 with the capabilities replaced by the 1000.

The HP 1000 includes many new software capabilities that are available to existing installations. Included is support for 2644A's and 2645A-007 CRT's with tape minicartridges as a system console plus on-line system generation and disc backup utilities for both the 5 and 15 megabyte discs. This new software is available with RTE-II and RTE-III and will be distributed in November to those installations signed up for the software subscription service. For other installations, the new software is available as upgrade kits. For RTE-II upgrades, order 92001B option 001 plus option 030 for 7900 5 Mbyte cartridges. For RTE-III upgrades, order 92060B option 001 plus either option 030 (7900) or 031 (7905). Both upgrades are available for $500.00 (U.S.).

NEW RELEASES FROM THE 2100/21MX CONTRIBUTED LIBRARY

Melanie Van Vliet
HP Data Systems Division

This article serves as an update for the 2100/21MX Contributed Library Program Catalog (22999-90040).

The new contributed programs listed below are now available. Contact your local HP sales office to order Contributed Library material, or (if you are in the U.S.) you can use the Direct Mail Order form at the back of the **COMMUNICATOR.**

22682-18921 FILE TRANSFER BETWEEN RTE AND DOS-M/III

This program provides the capability of transferring files between RTE and DOS-M or DOS-III systems. The program will

run under RTE-I, RTE-II, or RTE-III. A DOS-M or DOS-III disc must be placed into a peripheral disc subchannel. Files may then be transferred from RTE to the DOS-M/III disc, or vice versa. Source, relocatable, or binary files may be transferred. In the RTE host system, files handled may reside in the File Manager, on LS tracks, or on LG tracks. In the DOS-M/III system on the peripherals disc subchannel, files in the system or user areas may be handled.

Order Number	Option	Price
22682-18921	PT	$30.00
22682-13321	Cass	35.00

22682-18922 DOS-M/III FILE DIRECTORY LISTING FROM RTE ENVIRONMENT

This program provides the capability of listing the directory of a DOS-M or DOS-III disc pack from within an RTE environment. The program will run under RTE-I, RTE-II, or RTE-III. The DOS-M or DOS-III disc whose directory listing is desired is placed into a peripheral disc subchannel. The DOS-M/III directory listing generated by this program is identical with the directory listings generated by DOS-M or DOS-III. The system directory and user directory may both be listed.

Hardware requirements are RTE-I, II, or III and associated peripherals, at least 24k of core, and a disc drive having one or more RTE peripheral subchannels.

Order Number	Option	Price
22682-18922	PT	$20.00

22682-18923 DVR37-DOS DRIVER FOR ASCII BUS (HP 59310A)

This driver (DVR37) operates under the control of the I/O control module of the DOS execute.

This driver is responsible for:

1. Controlling input and output data transmission with the device(s) connected on the ASCII Bus Line via HP 59310A I/O card.
2. Outputting the Control Word which is used to select the operating modes and Bus functions that the Bus I/O card is to perform.

Order Number	Option	Price
22682-18923	PT	$25.00

22682-18924 DOS-III FILE SAVER

This program is capable of saving purged files and files whose boundaries are not the same in the user directory as they are actually on the disc.

This is a flexible file saving program which resides on disc within the Hewlett-Packard DOS-III system. The program when called into core is able to "save" purged files or files whose boundaries are not the same in the user directory as they are actually on the disc. For added protection of the user files, a security code is incorporated into the program along with a no echo input feature after the program has been called into core.

Order Number	Option	Price
22682-18924	PT	$10.00

22682-18925 TAPE-TO-TAPE COPY, SINGLE DRIVE MAG TAPE

A single drive mag tape copy utility for DOS-M and DOS-III. This program will copy HP FORTRAN generated tapes using only a single mag tape drive and the disc. It is a file oriented program that can take as input many files from many different tapes, in any order, and put them out on a fresh tape.

Order Number	Option	Price
22682-18925	PT	$10.00

22682-18926 UTDMP – UNIVERSAL MAG TAPE DUMP

This program provides a formatted dump of mag tape data. The data may be in any format and the block size may be of any practical length.

The program is conversational in obtaining the information needed for a job. (Dumps are on a per-file basis.)

The program gives both an octal and an alphanumeric representation of the data in a convenient form.

UTDMP – Main Program
RECRD – Subroutines

Order Number	Option	Price
22682-18926	PT	$10.00

22682-18927 **SIMPLEX – LINEAR PROGRAMMING FOR DOS-III**

This program solves linear programming problems by using the Simplex or Dantzig method. The program minimizes an objective function subject to up to 100 constraints with no more than 1000 variables (real + artificial + slack); the program needs as data the first tableaus of the method getting to the optimal solution.

The program can maximize problems if the objective function of the mathematical model is multiplied by -1.

Order Number	Option	Price
22682-18927	PT	$10.00

22682-10928 **APPLICATIONS DATA MANAGEMENT FOR HP RTE FMGR COMPUTER SYSTEMS (ADM-RTE)**

ADM is a file management and information retrieval system. The system has the following capabilities:

1. Record definition by data element.
2. File accessing methods;
   a. sequential
   b. binary search
   c. hash index
3. Record Retrieval by;
   a. relative record number
   b. key value
   c. conditional key value
   d. cross reference index
4. Report formatting
5. Data base utility functions
6. Sort

Order Number	Option	Price
22682-10928	800 BPI Mag Tape	$275.00
22682-11928	1600 BPI Mag Tape	275.00
22682-13328	Cass	325.00

22682-18929 **RTE DRIVER PACKAGE FOR XYNETICS & CALCOMP PLOTTERS**

This program package includes the RTE DRIVER FOR XYNETICS AND CALCOMP PLOTTERS (DVP10) and XYNETICS PLOTTER ROUTINES.

The driver allows Xynetics 1100, 1200, & 2000 series plotters equipped with SG-50 slope generators to operate as standard system peripheral equipment. Standard EXEC calls may be made to output data and check status of plotter. Standard system features such as buffered output, equipment timeout, etc., are available. This driver allows programmable pen color changes with automatic offsetting so that all colors line up. The driver also supports CALCOMP 560 series incremental plotters. Type of plotter to be used is specified by subchannel at system generation time. Plotter Routines (XYNETICS) – PLOT module replaces PLOT portion of standard HP plotter library. These routines format data for the XYNETICS flatbed plotters, mod 1100, 1200 & 2000 equipped with SG50 slope generators and HP interfaces. Each line drawn requires 8 words of data. Routine updates current pen coordinates, handles pen up/down commands, and new origin commands same as standard PLOT module. This is a utility subroutine, type 7, to be appended to users program which generates data. A test program is supplied to verify correct operation. The test program draws a simple geometric pattern using all four pen colors and plots a few symbols.

Hardware requirements – Xynetics Flatbed Plotter 1100, 1200, 2000 SG-50 slope generator C65A/21 HPIN interface card for HP 2100 computer P/N 102708. HP interface cable P/N 103018B. 4-pen in-line marker P/N WA40B. HP 12566B I/O card: W1 = A; W2 = C; W3 = B; W4 = B; W9 = A; W5-8 disconnect
HP RTE system.

Part Number	Option	Price
22682-18929	PT	$30.00
22682-13329	Cass	75.00

22682-18930 SYSTEM MAINTENANCE UTILITY (SMUT)

This routine allows, the RTE II or III user to perform useful disc and memory related functions for both 7900 and 7905 discs. The routine uses several system intrinsics, and can run in a 5k or 12k environment. It has a built in calculator for performing arithmetic and bit manipulation calculations. The following capabilities are offered by the routine:

1. display and alter memory
2. replace, copy, sum, and initialize blocks of memory
3. display and alter any disc
4. modify corresponding disc by address memory
5. list symbolic contents of track assignment table
6. list address of all system tables
7. list entire EQT, LU, INT, and IO tables
8. list contents of any programs ID segment
9. list compact table of HP-IB device talk/listen addresses
10. interrogate the BUS

A comprehensive users guide with numerous hints for effective utilization accompanies the program.

Order Number	Option	Price
22682-18930	PT	$30.00
22682-13330	CASS	70.00

WARNING – The main thrust of this program is to give the user full and complete control of an RTE system. Thus, some commands can have disasterous effects if not used correctly. The author and Hewlett-Packard will assume no responsibility for the effects of improper use.

22681-18932 RTE MULTIPLEXER DRIVER (12920A)

This routine is an Input/Output Driver (DVR52) which operates under control of the Real Time Executive (RTE) Operating system. Data transmission is possible with any EIA RS-232 compatible terminal connected to this interface. Speed detection is possible at speeds of 10, 15, 30, 60, 120 and 240 characters per second. This driver does not use the full capability of the Modem Control Card. This routine is a privileged driver and is compatible with the RTE, RTE-C, RTE-II, and RTE-III Operating Systems.

Order Number	Option	Price
22681-18932	PT	$20.00
22681-13332	Cass	35.00

22293-18900 OCTAL ASSEMBLY PROCESSOR & UTILITY SYSTEM (REV. C) (OCTAPUS-C)

Octapus-C performs all the functions of Octapus-B (Assembly and inverse assembly, paper tape punching, loading and verifying, and memory searching). In addition it provides the capability to handle EAU instructions, to reverse the skip sense of ASG instructions, to do an inverse assembly of a binary tape without loading it, and to relocate sections of core either on-line or by the production of a binary tape.

Order Number	Option	Price
22293-18900	PT	$20.00

The following programs have been "Withdrawn" from the Contributed Library. They are no longer available.

Documentation on these programs can be obtained only upon special request made to the 2100/21MX Contributed Library Manager.

ORDER NUMBER	TITLE
22682-10906 22682-11906	ON-TOP DEMO
22681-18963	TODS-C DISC EDITOR

The following programs have had revisions.

INTERIM TRAINING SCHEDULE

Jane Seligson
HP Data Systems

Because of the heavy demand for certain DS courses, the training schedule is being released prior to normal distribution in the Computer Systems Group Course Schedule. These dates are firm and reservations are being taken to fill them.

Also two of the most popular training manuals have been extensively redone and carry new part numbers. If you want one, use the new number. They are the RTE II/III Student Workbook (# 22999-90101), priced at $16.00, and Batch Spool Monitor (# 22999-90102) priced at $15.00.

software updates

Listed below are the software parts and manuals which are shipped when 92001B (RTE-II) is ordered (or is included in an order).

SOFTWARE MODULE NUMBERS: 92001B

Software Module Number	Module File Name	Module Descriptor	Part Numbers		
			Mini-Cartridge	7900 Disc	7905 Disc
02607-16004	!S4L07	24K SIO LINE PRINTER DRIVER	92001-13305	92001-13001	92001-13101
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13301	92001-13001	92001-13101
12970-16004	!S4MT1	24K SIO MAG. TAPE DRIVER	92001-13305	92001-13001	92001-13101
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13301	92001-13001	92001-13101
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13301	92001-13001	92001-13101
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13301	92001-13001	92001-13101
20875-60001	%1FTN	FORTRAN MAIN CONTROL	92060-13308	92001-13001	92001-13101
20875-60002	%2FTN	FORTRAN PASS 1	92060-13308	92001-13001	92001-13101
20875-60003	%3FTN	FORTRAN PASS 2	92060-13308	92001-13001	92001-13101
20875-60004	%4FTN	FORTRAN PASS 3	92060-13308	92001-13001	92001-13101
20875-60005	%5FTN	FORTRAN PASS 4	92060-13308	92001-13001	92001-13101
24129-60001	%ALGOL	RTE/DOS ALGOL PART 1	92060-13305	92001-13001	92001-13101
24129-60002	%ALGL1	RTE/DOS ALGOL PART 2	92060-13305	92001-13001	92001-13101
24153-60001	%FF.N	RTE/DOS FORMATTER	92060-13303	92001-13001	92001-13101
24170-60001	%1FTN4	RTE/DOS FORTRAN IV PART 1	92060-13306	92001-13001	92001-13101
24170-60002	%2FTN4	RTE/DOS FORTRAN IV PART 2	92060-13306	92001-13001	92001-13101
24170-60003	%3FTN4	RTE/DOS FORTRAN IV PART 3	92060-13306	92001-13001	92001-13101
24177-60001	%1FFT4	RTE/DOS FAST FORTRAN IV PART 1	92060-13307	92001-13001	92001-13101
24177-60002	%2FFT4	RTE/DOS FAST FORTRAN IV PART 2	92060-13307	92001-13001	92001-13101
24998-16001	%RLIB1	RTE/DOS LIBRARY PART 1	92060-13302	92001-13001	92001-13101
24998-16001	%RLIB2	RTE/DOS LIBRARY PART 2	92060-13302	92001-13001	92001-13101
24998-60002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	92001-13001	92001-13101
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13301	92001-13001	92001-13101
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13301	92001-13001	92001-13101
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13301	92001-13001	92001-13101
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13301	92001-13001	92001-13101
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13301	92001-13001	92001-13101
29100-60017	!S4LP	24K SIO LINE PRINTER	92001-13305	92001-13001	92001-13101
29100-60018	!S4SYD	24K SIO SYSTEM DUMP	92001-13305	92001-13001	92001-13101
29100-60019	!S4PHR	24K SIO PHOTO READER	92001-13305	92001-13001	92001-13101
29100-60020	!S4PUN	24K SIO TAPE PUNCH	92001-13305	92001-13001	92001-13101
29100-60022	!S4L67	24K SIO 2767 LINE PRINTER	92001-13305	92001-13001	92001-13101
29100-60023	!S4MT2	24K SIO 7970 MAG. TAPE	92001-13305	92001-13001	92001-13101
29100-60049	!S4MT3	24K SIO MAG. TAPE	92001-13305	92001-13001	92001-13101
29100-60050	!S4TER	24K SIO TERMINAL PRINTER	92001-13305	92001-13001	92001-13101
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13301	92001-13001	92001-13101
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13301	92001-13001	92001-13101
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13301	92001-13001	92001-13101
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13301	92001-13001	92001-13101
72009-60001	%2DV10	MIN. 7210A PLOTTER DRIVER	92062-13301	92001-13001	92001-13101
91200-16001	%DVA13	91200A DRIVER	92062-13301	92001-13301	92001-13101

Software Module Number	Module File Name	Module Descriptor	Part Numbers		
			Mini-Cartridge	7900 Disc	7905 Disc
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13301	92001-13001	92001-13101
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13301	92001-13001	92001-13101
92001-16002	%LDR2	RTE LOADER	92001-13301	92001-13001	92001-13101
92001-16003	%MTM	MULT. TERMINAL MONITOR	92001-13301	92001-13001	92001-13101
92001-16004	%2DP43	POWER FAILURE DRIVER	92001-13301	92001-13001	92001-13101
92001-16005	%SYLIB	RTE SYSTEM LIBRARY	92001-13301	92001-13001	92001-13101
92001-16012	%CR2SY	CORE RESIDENT OPERATING SYS.	92001-13301	92001-13001	92001-13101
92001-16013	!2GN00	RTE-II 7900 OFF-LINE GEN.	92001-13303	92001-13001	92001-13101
92001-16014	%AUTOR	AUTO RESTART PROGRAM	92001-13302	92001-13001	92001-13101
92001-16018	!2GNFH	RTE-II FIXED HEAD DISC GEN.	92001-13306	92001-13001	92001-13101
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13301	92001-13001	92001-13101
92001-16026	!2GN05	RTE-II 7905 OFF-LINE GEN.	92001-13303	92001-13001	92001-13101
92001-16027	%4DV05	RTE 2640A/2644A DRIVER	92062-13301	92001-13001	92001-13101
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13301	92001-13001	92001-13101
92001-16029	%$CMD2	RTE-II COMMAND PROGRAM	92001-13301	92001-13001	92001-13101
92001-16030	%WHZT2	RTE-II WHZAT PROGRAM	92001-13302	92001-13001	92001-13101
92001-16031	%RT2G1	RTE-II ON-LINE GENERATOR PART 1	92001-13304	92001-13001	92001-13101
92001-16031	%RT2G2	RTE-II ON-LINE GENERATOR PART 2	92001-13304	92001-13001	92001-13101
92001-18014	&AUTOR	AUTO RESTART SOURCE	92001-13302	92001-13001	92001-13101
92002-12001	%BMPG1	BATCH MONITOR PROGRAM PART 1	92002-13301	92001-13001	92001-13101
92002-12002	%BMPG2	BATCH MONITOR PROGRAM PART 2	92002-13301	92001-13001	92001-13101
92002-12001	%BMPG3	BATCH MONITOR PROGRAM PART 3	92002-13301	92001-13001	92001-13101
92002-12002	%2SPO1	RTE-II SPOOL MONITOR PART 1	92002-13303	92001-13001	92001-13101
92002-12002	%2SPO2	RTE-II SPOOL MONITOR PART 2	92002-13303	92001-13001	92001-13101
92002-16006	%BMLIB	BATCH LIBRARY	92002-13302	92001-13001	92001-13101
92002-16010	%EDITR	RTE EDITOR	92002-13302	92001-13001	92001-13101
92060-12004	%ASMB	RTE ASSEMBLER	92060-13304	92001-13001	92001-13101
92060-16028	%XREF	CROSS REFERENCE	92060-13304	92001-13001	92001-13101
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13301	92001-13001	92001-13101
92060-16038	%SWTCH	RTE-II SWITCH PROGRAM	92001-13304	92001-13001	92001-13101
92060-16039	%SAVE	SAVE PROGRAM	92060-13309	92001-13001	92001-13101
92060-16040	%RESTR	RESTORE PROGRAM	92060-13309	92001-13001	92001-13101
92060-16041	%VERFY	DISC VERIFY PROGRAM	92060-13309	92001-13001	92001-13101
92060-16042	%COPY	DISC COPY PROGRAM	92060-13309	92001-13001	92001-13101
92060-16043	%DBKLB	DISK BACK UP LIBRARY	92060-13309	92001-13001	92001-13101
92060-16044	!DSKBK	OFF-LINE DISK BACK UP	92060-13309	92001-13001	92001-13101
92060-16045	%RDNAM	READ NAMR PROGRAM	92001-13302	92001-13001	92001-13101
92060-18046	&UPDAT	UPDATE TRANSFER FILE	92001-13302	92001-13001	92001-13101
92060-18047	&PKDIS	PACK DISC TRANSFER FILE	92001-13302	92001-13001	92001-13101
92202-16001	%DVR23	RTE 7970 9T MAG. TAPE DRIVER	92062-13301	92001-13001	92001-13101

MANUAL NUMBERS: 92001B

Part Number	Part Description	Manual Type
92001-93003	RTE-II PART # CAT	Hardware
92001-93001 92060-90012	RTE-II MANUAL RTE NEW USERS GUIDE	Software Operating System
5951-1369 5951-1374 5951-1376 5951-1390	MAN–INTRODUCTION M-SIO SYS CONFG M-BBL/BBDL/BMDL MAN–STO SUBSYS.	Software Operating Procedures
02116-9015 02116-9072 24998-90001 5951-1321 92060-90005 92060-90017 92060-90020	MNL–FORTRAN MNL–ALGOL MNL–RTE/DOS LIBR MNL–FORTRAN IV ASSEMBLY MANUAL RTE UTILS MNL #1 ON-LINE GEN MNL 1	Language and Programming

Part Number	Part Description	Manual Type
02116-91760 02762-90002 12653-90004 12970-90901 12987-90006 13022-90010 13029-90010	M-2600 SIO PRGM M-2762/2615 SIO MNL-SIO DR MANUAL MNL-SIO LP DRVR M-7970B SIO DR MNL-7970 7T SIO	Small Programs
92060-90010	MNL POCKET GUIDE	Pocket Guide

Listed below are the software parts and manuals which are shipped when 92060B (RTE-III) is ordered (or is included in an order).

SOFTWARE MODULE NUMBERS: 92060B

Software Module Number	Module File Name	Module Descriptor	Part Numbers		
			Mini-Cartridge	7900 Disc	7905 Disc
02607-16004	!S4L07	24K SIO LINE PRINTER DRIVER	92001-13305	92060-13001	92060-13101
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13301	92060-13001	92060-13101
12970-16004	!S4MT1	24K SIO MAG. TAPE DRIVER	92001-13305	92060-13001	92060-13101
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13301	92060-13001	92060-13101
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13301	92060-13001	92060-13101
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13301	92060-13001	92060-13101
20875-60001	%1FTN	FORTRAN MAIN CONTROL	92060-13308	92060-13001	92060-13101
20875-60002	%2FTN	FORTRAN PASS 1	92060-13308	92060-13001	92060-13101
20875-60003	%3FTN	FORTRAN PASS 2	92060-13308	92060-13001	92060-13101
20875-60004	%4FTN	FORTRAN PASS 3	92060-13308	92060-13001	92060-13101
20875-60005	%5FTN	FORTRAN PASS 4	92060-13308	92060-13001	92060-13101
24129-60001	%ALGOL	RTE/DOS ALGOL PART 1	92060-13305	92060-13001	92060-13101
24129-60002	%ALGL1	RTE/DOS ALGOL PART 2	92060-13305	92060-13001	92060-13101
24153-60001	%FF.N	RTE/DOS FORMATTER	92060-13303	92060-13001	92060-13101
24170-60001	%1FTN4	RTE/DOS FORTRAN IV PART 1	92060-13306	92060-13001	92060-13101
24170-60002	%2FTN4	RTE/DOS FORTRAN IV PART 2	92060-13306	92060-13001	92060-13101
24170-60003	%3FTN4	RTE/DOS FORTRAN IV PART 3	92060-13306	92060-13001	92060-13101
24177-60001	%1FFT4	RTE/DOS FAST FORTRAN IV PART 1	92060-13307	92060-13001	92060-13101
24177-60002	%2FFT4	RTE/DOS FAST FORTRAN IV PART 2	92060-13307	92060-13001	92060-13101
24998-16001	%RLIB1	RTE/DOS LIBRARY PART 1	92060-13302	92060-13001	92060-13101
24998-16001	%RLIB2	RTE/DOS LIBRARY PART 2	92060-13302	92060-13001	92060-13101
24998-16002	%FF4.N	FORTRAN IV FORMATTER	92060-13303	92060-13001	92060-13101
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13301	92060-13001	92060-13101
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13301	92060-13001	92060-13101
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13301	92060-13001	92060-13101
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13301	92060-13001	92060-13101
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13301	29060-13001	92060-13101
29100-60017	!S4LP	24K SIO LINE PRINTER	92001-13305	92060-13001	92060-13101
29100-60018	!S4SYD	24K SIO SYSTEM DUMP	92001-13305	92060-13001	92060-13101
29100-60019	!S4PHR	24K SIO PHOTO READER	92001-13305	92060-13001	92060-13101
29100-60020	!S4PUN	24K SIO TAPE PUNCH	92001-13305	92060-13001	92060-13101
29100-60022	!S4L67	24K SIO 2767 LINE PRINTER	92001-13005	92060-13001	92060-13101
29100-60023	!S4MT2	24K SIO 7970 MAG. TAPE	92001-13305	92060-13001	92060-13101
29100-60049	!S4MT3	24K SIO MAG. TAPE	92001-13305	9206 13001	92060-13101
29100-60050	!S4TER	24K SIO TERMINAL PRINTER	92001-13305	92060-13001	92060-13101
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13301	92060-13001	92060-13101
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13301	92060-13001	92060-13101
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13301	92060-13001	92060-13101
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13301	92060-13001	92060-13101
72009-60001	%2DV10	MIN. 7210A PLOTTER DRIVER	92062-13301	92060-13001	92060-13101
91200-16001	%DVA13	91200A DRIVER	92062-13301	92060-13001	92060-13101
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13301	92060-13001	92060-13101
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13301	92060-13001	92060-13101
92001-16003	%MTM	MULT. TERMINAL MONITOR	92060-13301	92060-13001	92060-13101

Software Module Number	Module File Name	Module Descriptor	Part Numbers		
			Mini-Cartridge	7900 Disc	7905 Disc
92001-16005	%SYLIB	RTE SYSTEM LIBRARY	92060-13301	92060-13001	92060-13101
92001-16014	%AUTOR	AUTO RESTART PROGRAM	92060-13310	92060-13001	92060-13101
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13301	92060-13001	92060-13101
92001-16027	%4DV05	RTE 2640A/2644A DRIVER	92062-13301	92060-13001	92060-13101
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13301	92060-13001	92060-13101
92001-18014	&AUTOR	AUTO RESTART PROGRAM SOURCE	92060-13310	92060-13001	92060-13101
92002-12001	%BMPG1	BATCH MONITOR PROGRAM PART 1	92002-13301	92060-13001	92060-13101
92002-12001	%BMPG2	BATCH MONITOR PROGRAM PART 2	92002-13301	92060-13001	92060-13101
92002-12001	%BMPG3	BATCH MONITOR PROGRAM PART 3	92002-13301	92060-13001	92060-13101
92002-16006	%BMLIB	BATCH LIBRARY	92002-13302	92060-13001	92060-13101
92002-16010	%EDITR	RTE EDITOR	92002-13302	92060-13001	92060-13101
92060-12001	%3SPO1	RTE-III SPOOL MONITOR PART 1	92060-13313	92060-13001	92060-13101
92060-12001	%3SPO2	RTE-III SPOOL MONITOR PART 2	92060-13313	92060-13001	92060-13101
92060-12003	%CR3SY	MEMORY RESIDENT SYSTEM	92060-13301	92060-13001	92060-13101
92060-12004	%ASMB	RTE ASSEMBLER	92060-13304	92060-13001	92060-13101
92060-16001	%3DP43	POWER FAILURE DRIVER	92060-13301	92060-13001	92060-13101
92060-16004	%LDR3	RTE-III LOADER	92060-13301	92060-13001	92060-13101
92060-16006	%WHZT3	RTE-III WHZAT PROGRAM	92060-13310	92060-13001	92060-13101
92060-16028	%XREF	CROSS REFERENCE	92060-13304	92060-13001	92060-13101
92060-16029	!3GN00	7900 RTE-III GENERATOR	92060-13311	92060-13001	92060-13101
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13301	92060-13001	92060-13101
92060-16032	!3GN05	7905 RTE-III GENERATOR	92060-13311	92060-13001	92060-13101
92060-16035	%$PVMP	$PVMP	92060-13301	92060-13001	92060-13101
92060-16036	%$CMD3	RTE-III COMMAND PROGRAM	92060-13301	92060-13001	92060-13101
92060-16037	%RT3G1	RTE-III ON-LINE GENERATOR PART 1	92060-13312	92060-13001	92060-13101
92060-16037	%RT3G2	RTE-III ON-LINE GENERATOR PART 2	92060-13312	92060-13001	92060-13101
92060-16038	%SWTCH	RTE-III SWITCH PROGRAM	92060-13312	92060-13001	92060-13101
92060-16039	%SAVE	SAVE PROGRAM	92060-13309	92060-13001	92060-13101
92060-16040	%RESTR	RESTORE PROGRAM	92060-13309	92060-13001	92060-13101
92060-16041	%VERFY	DISC VERIFY PROGRAM	92060-13309	92060-13001	92060-13101
92060-16042	%COPY	DISC COPY PROGRAM	92060-13309	92060-13001	92060-13101
92060-16043	%DBKLB	DISK BACK UP LIBRARY	92060-13309	92060-13001	92060-13101
92060-16044	!DSKBK	OFF LINE DISK BACK UP	92060-13309	92060-13001	92060-13101
92060-16045	%RDNAM	READ NAMR PROGRAM	92060-13310	92060-13001	92060-13101
92060-18046	&UPDAT	UPDATE TRANSFER FILE	92060-13310	92060-13001	92060-13101
92060-18047	&PKDIS	PACK DISK TRANSFER FILE	92060-13310	92060-13001	92060-13101
92202-16001	%DVR23	RTE 7470 9T MAG. TAPE DRIVER	92062-13301	92060-13001	92060-13101

MANUAL NUMBERS: 92060B

Part Number	Part Description	Manual Type
92060-90004 92060-90012	RTE-III MANUAL RTE NEW USERS GUIDE	Software Operating System
5951-1369 5951-1374 5951-1376 5951-1390	MAN–INTRODUCTION M-SIO SYS CONFG M-BBL/BBDL/BMDL MAN–SIO SUBSYS.	Software Operating Procedures
02116-9015 02116-9072 24998-90001 5951-1321 92060-90005 92060-90017 92060-90020	MNL–FORTRAN MNL–ALGOL MNL–RTE/DOS LIBR MNL–FORTRAN IV ASSEMBLY MANUAL RTE UTILS MNL #1 ON-LINE GEN MNL 1	Language and Programming
02116-91760 02762-90002 12653-90004 12970-90901 12987-90006	M-2600 SIO PRGM M-2762/2615 SIO MNL–SIO DR MANUAL MNL–SIO LP DRVR	Small Programs

Part Number	Part Description	Manual Type
13022-90010 13029-90010	M-7970B SIO DR MNL—7970 7T SIO	Small Programs Continued
92060-90019	RTE III PT # CAT	Miscellaneous
92060-90010	MNL POCKET GUIDE	Pocket Guide

RTE DRIVERS

Following is a list of the drivers available for RTE systems.

RTE DRIVERS

Software Module Number	Module File Name	Module Descriptor	Part Numbers		
			Mini-Cartridge	7900 Disc	7905 Disc
09601-16021	%DVR15	RTE 7261A DRIVER	92062-13301		
20747-60001	%DVR30	RTE FIXED HEAD DISC DRIVER	92062-13301		
20808-60001	%CAL10	CAL. PLOTTER DRIVER	92062-13301		
20810-60001	%CALIB	CAL. PLOTTER LIBRARY	92062-13301		
25117-60499	%DVR24	RTE 7970 7T MAG. TAPE DRIVER	92062-13301		
29013-60001	%DVR31	RTE 7900A DISC DRIVER	92062-13301		
29028-60002	%DVR12	RTE 2767A DRIVER	92062-13301		
29029-60001	%DVR00	RTE TTY/PUNCH/PHOTO READER	92062-13301		
29030-60001	%DVR11	RTE 2892A CARD READER DRIVER	92062-13301		
59310-16002	%1DV37	RTE HP-IB WITHOUT SRQ	92062-13301		
59310-16003	%2DV37	RTE HP-IB WITH SRQ	92062-13301		
59310-16004	%HPIB	HP-IB DEVICE SUBROUTINE	92062-13301		
72008-60001	%1DV10	COMP. 7210A PLOTTER DRIVER	92062-13301		
72009-60001	%2DV10	MIN. 7210A PLOTTER DRIVER	92062-13301		
91200-16001	%DVA13	91200A DRIVER	92062-13301		
91200-16002	%TVLIB	91200A VIDEO MONITOR LIBRARY	92062-13301		
91200-16004	%TVVER	91200A TV INTERFACE VERIFIER	92062-13301		
92001-16020	%DVA12	2607/10/13/14/17/18 DRIVER	92062-13301		
92001-16027	%4DV05	RTE 2640A/2644A DRIVER	92062-13301		
92001-16028	%0DV05	RTE 2640A DRIVER	92062-13301		
92060-16031	%DVR32	RTE 7905A DISC DRIVER	92062-13301		
92202-16001	%DVR23	RTE 7470 9T MAG. TAPE DRIVER	92062-13301		

DOS-IIIB MODULES

The Index below indicates the modules available for DOS-IIIB systems, HP 24307B, date code 1523.

This Index relates the names of the relocatable modules to the part numbers of the equivalent paper tapes and indicates the purpose of the modules. Modules not specifically designated for the 2100A/S or for the 21MX computers are to be used on either.

NAME	PART NUMBER	REV	DESCRIPTION
DISCH	24307-16069	1523	DISC MONITOR
$EXMD	24307-16070	1523	EXEC MODULES
DVR00	20985-60001	1516	TTY-LIKE CONSOLE/ TERMINAL
DVR01	20987-60001	1419	PAPER TAPE READER
DVR02	20989-60001	1419	PAPER TAPE PUNCH
DVR05	24157-60001	1419	TTY-LIKE CONSOLE
DVR15	24307-16017	1446	7261A MARK SENSE CARD READER
D2892	24272-60001	1419	2892B CARD READER (DVR11)
D2767	24168-60001	1419	2767A LINE PRINTER (DVR12)
D26XX	24307-16011	1446	DVR12 FOR 2607, 2610, 2614, 2613, 2618
DVR23	13024-60001	1446	7970B/E MAG TAPE
DVR26	24307-16018	1507	2762A/B AND 2615A CONSOLE
DVR30	24307-16073	1523	DISC BATCH DRIVER
DVR31	24156-60001	1419	7900/7901 DISC
DVR67	24341-60001	1419	12889A HI SPD SERIAL IF
DVR70	24307-16009	1446	DVR70 FOR 12618A SYNC INTERFACE
DVR71	24307-16013	1515	12967A SYNC MODEM IF
DVR72	24350-16001	1523	12587B ASYNC DATA SET IF

documentation

NAME	PART NUMBER	REV	DESCRIPTION
DVR73	24377-16001	1523	12920A/B MULTIPLEXOR
DVR74	24307-16014	1515	12966A/12968A ASYNCH IF
EFMP	24309-60002	1523	EXT FILE MGR EXEC MODULES
	24309-60003	1523	EXT FILE MGR UTILITIES
JOBPR	24307-16071	1523	JOB PROCESSOR
RLODR	24307-16072	1523	RELOCATING/LINKING LOADER
ASMB	24307-16006	1419	2100/21MX ASSEMBLER
.FTN4	24170-60001	C	FORTRAN IV COMPILER
	24170-60002	C	
	24170-60003	C	
FTN4	24177-60001	1442	FORTRAN IV COMPILER (10K AREA)
	-60002	1442	
ALGOL	24129-60001	C	ALGOL COMPILER
	24129-60002	C	
XREF	24223-60001	1523	2100/21MX CROSS REF TABLE GEN
F4D.N	24152-60001	C	RELO SUBRLIBR FTN4
F2E.N	24151-60001	D	RELO SUBR LIBR (EAU)
F2F.N	24248-60001	B	RELO SUBR LIBR (FP)
FFP.N	12907-16001	A	2100A/S FFP SUBR LIBRARY
$SETP	12907-16002	1350	2100A/S FFP SUBR $SETP
ATD01	24381-16001	1503	ASYNC TERMINAL DRIVER No. 1
ATD02	24307-16012	1442	ASYNC TERMINALL DRIVER No. 2
PMT01	24307-16008	1438	PAGE MODE TERMINAL DRIVER No. 1
PMT02	24307-16016	1503	PAGE MODE TERMINAL DRIVER No. 2
SLC	24307-16010	1438	SYNCHRONOUS LINE CONTROL DRIVER
DVR33	24278-60001	1419	2100/21MX WCS DRIVER
MASMB	24332-60001	1419	2100A/S WCS MICRO ASSEMBLER
WCSUT	24333-60001	A	2100/21MX MICRO UTILITIES
MDBUG	24334-60001	1419	2100A/S WCS MICRO DEBUG EDITR
XASMB	12978-16001	1437	21MX WCS MICRO ASSEMBLER
XDBUG	12978-16002	1437	21MX WCS MICRO DEBUG EDITOR
FFP.X	12977-16001	1451	21MX FFP SUBR LIBRARY
XSETP	12977-16002	1451	21MX FFP SUBR $SETP

The following tables list currently available customer manuals for Data Systems Division products. This list supersedes the list in the last issue of the **Communicator.**

The most recent changes to the tables are indicated for easy reference. Prices are subject to change without notice.

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package only, send your request to:

Software/Publications Distribution
11000 Wolfe Road
Cupertino, Ca. 95014

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manual(s) you need on the Corporate Parts Center form supplied at the back of the **Communicator**.

A few words about documentation terms:

New — A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number.

Revised — A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.

Update — An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number; it is automatically included when you order the manual with which it is associated.

9600/9700 SYSTEMS MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02313-93002	RTE 2313B Analog-Digital Interface Subsystem Operating and Service Manual	$30.00	8/76	
02320-93002	RTE System Driver DVR76 for HP 2320A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
02321-93001	RTE System Driver DVR74 for HP 2321A Low Speed Data Acquisition Subsystem Programming and Operating Manual	1.00	8/74	
09600-93010	RTE System DVR11 for HP 2892A Card Reader Programming and Operating Manual	1.00	8/74	
09600-93015	91200A TV Interface Kit; Programming and Operating Manual	4.50	7/75	1/76
09601-93007	RTE Device Subroutine for HP 5327A/B-H48 Counter	2.50	12/74	
09601-93009	RTE Device Subroutine for HP 5326A-H18 Counter	2.50	12/74	
09601-93014	RTE System Driver DVR15 Mark Sense Card Reader Programming and Operating Manual	1.00	2/76*O	
09601-93015	RTE for 40-bit Output Register #12556B	1.00	10/74	
09603-93001	9603A/9604A Control System and Scientific Measurement Operating and Service Manual	7.50	5/76	
09610-93003	ISA FORTRAN Extension Package Reference Manual	4.50	7/76*R	
09611-90009	9611A Operating 406 Industrial Measurement and Control System	.25	4/75	
09611-90010	HP 6940A/B Multiprogrammer Verification Manual	4.50	8/75	
12604-93002	RTE DVR40 for 12604B Data Source Interface	1.00	8/74	
12665-93001	RTE System Driver DVR65 for HP 12771A Computer Serial Interface Kit	1.00	8/74	
12989-99001	RTE System Driver DVA15 for Card Reader Punch Subsystem 2894	1.00	1/75	5/76
24998-90001	DOS/RTE Relocatable Library Reference Manual	10.00	3/76	
25117-93003	RTE System Driver DVR24 for HP 7970 Series Digital Magnetic Tape Unit	1.00	8/74	
29003-93001	RTE System Driver DVR66 for HP 12772A Coupler Modem Interface Kit Programming and Operating Manual	1.00	8/74	
29003-93003	RTE System Driver DVR66 for HP 12770A Coupler Serial Interface Kit Programming and Operating Manual	1.00	8/74	
29009-93001	RTE System Driver DVR62 for HP 2313B Subsystem	2.50	8/74	
29013-90001	DVR31 RTE Moving Head Driver	10.00	2/73	
29014-90001	Moving Head Real-Time System Generator	20.00	4/72*O	
29015-90001	Fixed Head Real-Time System Generator	15.00	4/72	
29016-90002	RTE Scheduler	50.00	9/72	
29016-90003	Real-Time Input/Output Control	50.00	12/73	
29022-90001	Real-Time Relocating Loader	10.00	6/73*O	
29028-95001	RTE HP 2610A/2614A Line Printer Driver	1.50	8/73	
29029-91001	Real-Time Executive Multiple-Device System Control Device (DVR00) Program Listing	10.00	9/72	
29029-95001	Real-Time Executive System Driver DVR00 for Multiple Device System Control Small Programs Manual	1.50	11/75	
29033-98000	Real-Time Executive-File Manager System	10.00	3/73*O	
29100-93001	RTE System Driver DVR40 (29100-60041) for HP 12604B Data Source Interface Programming and Operating Manual	1.00	8/76*R	
29100-93003	RTE System Driver DVR61 for HP 6940A, 6941A Bidirectional Multiprogrammer Programming and Operating Manual	4.50	3/76	

*O = Obsolete Manual *R = Revised Manual

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
29101-93001	RTE Core-Based Software System Users Manual	$10.00	1/76*R	
29102-93001	RTE BASIC Software System Programming and Operating Manual	10.00	3/74	8/75
29103-93001	RTE System Cross Loader; Programming and Operating Manual	2.50	3/75	11/75
91060-93005	RTE Driver for X-Y Display Storage Subsystem (HP Model 1331C-016) Programming and Operating Manual	1.00	8/74	
91062-93003	Real-Time Executive System Driver for DVM/Scanner Subsystem	9.00	8/74	
92001-90015	RTE DVR05 for 264X Terminals	2.00	9/76*R	
92001-93001	RTE-II Software System Programming and Operating Manual	10.00	7/76*R	5/76
92060-90004	RTE-III Software System Programming and Operating Manual	12.00	7/76*R	
92060-90005	RTE Assembler Reference Manual	7.00	1/76	
92060-90009	RTE-III General Information Manual	4.00	2/76	
92060-90010	RTE Batch/Spool Monitor and Operating System Pocket Guide	3.00	10/75	
92060-90012	RTE: A Guide for New Users	6.50	7/76*R	
92060-90013	Batch-Spool Monitor Reference Manual	9.50	7/76*R	
92060-90014	RTE Interactive Editor Reference Manual	6.00	3/76	
92060-90016	Multi-User Real-Time BASIC Reference Manual	12.00	10/75	12/75
92060-90017	RTE Utility Programs	3.00	7/76*N	
92060-90020	RTE On-Line Generator	15.00	7/76*N	9/76
92063-90001	IMAGE/1000 Data Base Management System Reference Manual	9.00	6/76	
92200-93001	RTE System Driver DVR12 for HP 2607A Line Printer Programming and Operating Manual	1.00	3/74	
92200-93005	Real-Time Executive Operating System Drivers and Device Subroutine Manual	5.00	7/76*R	
92202-93001	RTE System Driver DVR23 for HP 7970 Series Digital Mag Tape Units Programming and Operating Manual	1.00	8/74	
93005-93005	Thermal Line Printer Subsystem for Driver DVR00 (RTE)	2.50	12/74	
93513-90002	RTE System Driver DVA76-DVR40 for 2801 Quartz Thermometer System	1.50	4/75	

SOFTWARE NUMBERING CATALOG MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
92060-90019	92060B Software Numbering Catalog	$ 1.50	7/76*N	
92001-93003	92001B Software Numbering Catalog	1.50	7/76*N	
92002-93003	92002A Software Numbering Catalog	1.50	7/76*N	
91705-93003	91705A Software Numbering Catalog	1.00	7/76*N	
92413-90001	92413A Software Numbering Catalog	1.00	7/76*N	
91780-93004	91780A Software Numbering Catalog	1.00	7/76*N	
92062-90001	92062A Software Numbering Catalog	1.00	7/76*N	
92101-90001	92101A Software Numbering Catalog	1.00	7/76*N	
92409-93002	92409A Software Numbering Catalog	1.00	7/76*N	
92400-93003	92400A Software Numbering Catalog	1.00	7/76*N	
91703-93003	91703A Software Numbering Catalog	1.00	7/76*N	
91704-93003	91704A Software Numbering Catalog	1.00	7/76*N	
91700-93005	91700A Software Numbering Catalog	1.00	7/76*N	
92063-90003	92063A Software Numbering Catalog	1.00	9/76*N	
92066-90001	92066A Software Numbering Catalog	1.00	9/76*N	

*N = New Manual R = Revised Manual

SOFTWARE INPUT/OUTPUT SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02116-91760	Teleprinter Driver (LP Compatible) Manual	$ 1.00	8/73	
02762-90002	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90003	HP 2892A Card Reader Driver	1.50	6/72	
12602-90022	Mark Sense Card Reader Drivers	1.00	6/70	
12653-90004	HP 2767 Line Printer Driver	1.00	9/70	1/73
12845-90005	HP 2610A/2614A Line Printer Driver	1.00	2/74	
12987-90006	HP 2607 Line Printer Driver	5.00	11/73	
13022-90010	HP 7970 Magnetic Tape Unit Driver	1.00	2/72	
13029-90010	Magnetic Tape Driver (7-Track)	1.00	2/72	
5950-9276	SIO Drum-Disc	1.00	2/70	
5951-1374	Software Input/Output System Configuration	1.00	7/74	
5951-1390	Subsystem Operation	2.00	2/76	
59310-90063	RTE DVR37 459310 B Interface Bus Programming and Operating Manual	3.50	5/76	

BASIC CONTROL SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02022-90014	Magnetic Tape Reformatting System Support Utilities	$ 1.50	1/74	
02100-90129	HP 2100 Microassembler Coding Form	3.00		
02100-90140	Decimal String Arithmetic Routines	5.00	10/73	
02100-90200	Loader/Loader Reference Manual	2.00	5/76	
02108-90008	Microprogramming 21MX Computers Reference Manual	6.50	2/76	5/76
02116-9017	Basic Control System Manual	8.50	12/71	
02116-9072	ALGOL for HP 2000 Computers Reference Manual	10.00	2/76* R	
02116-91751	Prepare Tape System	2.50	8/74	
02116-91752	Magnetic Tape System	6.00	6/71	
02116-91780	2100 Series Relocatable Subroutines	11.00	12/74	
02762-90003	HP 2762A Terminal Printer Driver	1.00	5/73	
02892-90004	HP 2892A Card Reader Driver	1.50	6/72	
12602-90021	Mark Sense Drivers	1.00	6/70	
12653-90005	HP 2767 Line Printer Driver	1.00	10/70	
12845-90004	HP 2610A/2614A Line Printer Driver	1.00	6/72	
12987-90008	HP 2607 Line Printer Driver	5.00	12/73	
13023-90010	HP 7970 Magnetic Tape Unit Driver	1.00	5/74	
13026-90010	Magnetic Tape Driver (7-Track without DMA)	1.00	5/71	6/72
13027-90010	Magnetic Tape Driver (7-Track with DMA)	1.00	5/71	6/72
5951-1371	HP 2100 Front Panel Procedures	1.00	8/73	
5951-1376	Basic Binary Loader/Disc Loader, Basic Moving-Head Disc Loader	2.50	4/76	
5951-1391	Basic Control System	1.50	10/74	
5951-1392	Magnetic Tape System	1.00	7/71	

*R = Revised Manual

DISC OPERATING SYSTEM MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02767-00007	DOS/RTE 2767 Line Printer Driver	$ 1.00	12/70	
12560-90023	DOS RTE and BCS Calcomp Plotter Drivers	1.50	10/75	
12602-90023	DOS/RTE Mark Sense Drivers Kit 12602B	1.00	8/70	
12908-90004	HP 12908 Writable Control Store Driver	1.00	2/75	
24307-90006	DOS-III Reference Manual	20.00	1/76	3/76
24307-90012	DOS-III Data Communications Drivers	7.50	4/76	
24307-90018	DOS-III Pocket Guide	3.50	12/75	
24307-90022	DOS-III Terminal Printer Driver	1.00	1/75	
24307-90073	DOS-III Standard Drivers	6.00	1/75	
24376-90001	IMAGE/2000 Data Base Management System Reference Manual	11.00	8/75	
5951-1366	Cross Reference Table Generator	1.00	8/74	
5951-1381	DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
5951-1394	2000C File Interface for DOS-M	1.00	6/71	

LANGUAGE MANUALS

PART NUMBER	MANUAL TITLE	PRICE	DATE	UPDATE
02116-9014	HP Assembler Manual	$ 6.50	8/75	
02116-9015	HP FORTRAN Manual	5.00	3/74	
02116-9016	Symbolic Editor	4.50	2/74	
02116-9072	ALGOL Reference Manual	10.00	2/76	
12907-90010	Implementing the HP 2100 Fast FORTRAN Processor	1.00	7/76*R	
24307-90014	DOS-III Assembler Reference Manual	8.00	7/74	11/75
92060-90005	RTE Assembler Reference Manual	7.00	1/76	
5951-1321	HP FORTRAN IV Reference Manual	6.00	12/75	

*R = Revised Manual

training schedule

The schedule for customer training courses on Data Systems Division products has been expanded to include courses offered at our European training centers. Listed below are courses offered in the U.S. and in Europe during the period November, 1976 through April, 1977.

You can also obtain a copy of the training schedule from your local HP sales office. A European course schedule is available through the sales offices in Europe; a U.S. schedule through U.S. sales offices.

*Prices quoted are for courses at the two U.S. training centers only. For prices of courses at European training centers please consult your local HP Sales Office.

Registration

Requests for enrollment in any of the above courses should be made through your local HP representative. He will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the Training Course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time of registration.

Cancellations

In the event you are unable to attend a class for which you are registered please notify the Training Center Registrar immediately in order that we may offer your seat to another student.

Training Center Addresses

Cupertino
11000 Wolfe Road
Cupertino, California 95014
(408) 257-7000

Rockville
4 Choke Cherry Road
Rockville, Maryland 20850
(301) 948-6370

Boise
P. O. Box 15
15 N. Phillippi Street
Boise, Idaho 83707
(208) 376-6000
TWX: 910-970-5784

Boblingen
Kundenschulung
Herrenbergerstrasse 110
D-7030 Böblingen, Wurttemberg
Tel: (07031) 667-1
Telex: 07265739
Cable: HEPAG

Winnersh
King Street Lane
GB-Winnersh, Wokingham
Berks RG11 5 AR.
Tel: Wokingham 784774
Cable: Hewpie London
Telex: 847178 9

Grenoble
5, avenue Raymond-Chanas
38320 Eybens
Tel: (76) 25-81-41
Telex: 980124

Milan
Via Amerigo Vespucci, 2
1-20124 Milan
Tel: (2) 62 51
Cable: HEWPACKIT Milano
Telex: 32046

Madrid
Jerez No 3
E-Madrid 16
Tel: (1) 458 26 00
Telex: 23515 hpe

Stockholm
Enighetsvägen 1-3, Fack
S-161 20 Bromma 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS
Stockholm
Telex: 10721

TITLE			TRAINING COURSE RATES AND CENTER LOCATION									
Course Number	Length	Price	Cupertino	Rockville	Boise	Boblingen	Winnersh	Grenoble	Milan	Madrid	Stockholm	Amsterdam/ Brus.
22940A	2100 Maint.		1/17 3/14 4/11									
	10 days	$1000										
22941A	21MX Maint.		12/6 1/3 2/28									
	5 days	500										
22942A	7900 Maint.		12/13 1/10 3/28									11/22
	5 days	500										
22943A	7970B Maint.				3/7 4/25							11/29
	5 days	600										
22944A	7970E Maint.				1/31 4/18							
	5 days	600										
22945A	7905 Maint.		11/1 1/31 3/7 4/25									12/13
	5 days	500										
22950A	2100 Ser. Assm.		11/15 12/6 1/3 1/31 2/28	11/1 11/29 1/17 1/31 3/14 4/25		12/13	12/6	11/22	11/29		11/22	11/22
	5 days	500										
22952A	DOS III B		11/1									12/13
	5 days	500										

Course Number	Length	Price	Cupertino	Rockville	Boise	Boblingen	Winnersh	Grenoble	Milan	Madrid	Stockholm	Amsterdam/ Brus.
22953A	2100 Image											
	3 days	300										
22959A	Assembler/21MX		12/6 1/3 1/31 2/28 3/21 4/11									
	5 days	500										
22960A	21MX Mic. Prog.		12/13 2/7 3/28									
	5 days	500										
22965B	RTE-II/III		11/8 11/29 *{1/3 1/10} *{1/10 1/17} *{1/17 1/24} *{1/31 2/7} *{2/7 2/14} 2/28	11/8 12/6 1/3 1/24 2/7 2/28 3/21 4/11		*{11/22 11/29}	*{1/10 1/17}	*{11/29 12/6} *{12/6 12/13}			*{11/22 12/6}	
	10 days	1000										
	(Course includes RTE-II/III operating system, batch spool monitor and file manager.)											
22968A	Measurement & control		12/13	12/20								
	2 days	200										
22969A	Distb. Sys.		12/13 1/17 2/14 3/28	3/14				11/15				
	5 days	500										
22977A	Image/DBMS 1000		11/8 1/24 3/21 4/25	1/17 4/25								
	5 days	500										
22978	TCS		11/15									
	2 days	200										
22979A	Real/Time Multiterminal Basic		12/15 2/23 4/4	11/22							12/13	
	3 days	300										
22980A	HPIB Multicomputer Bus Basic		1/24 3/14 4/25									
	3 days	300										
22981A	HPIB Programming Under RTE		1/27 3/17 4/28									
	2 days	200										

NOTE: Dates within brackets are starting dates for week 1 and week 2 of the RTE course. In some cases there is a break between the two weeks of the class. Course 22977A, IMAGE/DBMS 1000 replaces 22953A (2100 IMAGE); the new class adds additional material and extends the training from 3 to 5 days.

introducing the HP 1000

Figure 1. The new HP 1000 Model 30 system

The introduction of the Hewlett-Packard 1000 Computer System is one of those fortunate occasions where a number of computer advances arrive concurrently and are able to be utilized within a single new computer system. Significant contributions embodied within the HP 1000 computer system include a very fast new processor, and a fast and flexible new CRT terminal that serves as the system console and convenient machine/human interface to HP's new IMAGE/1000 data base management software. Plus, the system's contemporary and attractive desk styling is a welcome addition to office environments.

Beyond these new capabilities, the HP 1000 builds on previous contributions such as HP's complete computer network software, the Hewlett-Packard Interface Bus (HP-IB) for control of automated instrument systems, HP's efficient and proven Real Time operating systems, and the fastest cartridge disc memory on the market.

The effect is an exceptionally fast and powerful small computer system that both OEM's and end users with computer experience can easily use as a tool to implement a wide range of applications. The HP 1000 is especially well suited to computation, instrumentation, and operations management applications that demand high performance. And because its increased performance is priced five to ten per cent below previous HP small computer systems, the HP 1000 sets a new price/performance standard in its class. Figures 1 and 2 are the HP 1000 model 30 and 80 systems.

Figure 2. The new HP 1000 Model 80 system

DATA BASE MANAGEMENT WITH THE HP 1000

The combination of an HP 1000 and IMAGE/1000, Hewlett-Packard's new data base management software, provides a new entry level for data base management applications. For $61,200 Hewlett-Packard provides a complete set of tools that allows better management of manufacturing and design information. IMAGE/1000 assists in organizing a company's individual data files into a single data base so that relevant information is available to those who need it, when they need it. IMAGE/1000 utilizes a logical and easy-to understand structure that makes the construction of a data base a straight-forward task. IMAGE automatically links together related information and reduces redundant data. Users can have multi-terminal, multi-program access to the data base for concurrent and interactive retrieval and reporting of information.

Using IMAGE/1000 doesn't require programming skills. Non-technical people can access the data base with QUERY, a "free form" inquiry language. By merely typing simple, English-like commands on a terminal, an authorized operator can retrieve, enter, modify or delete data.

With QUERY, a user can quickly find any record in the data base by a "key value", such as a name, customer account or part number. There's no need to know the address of the data, or to search sequentially through record after record.

QUERY is especially useful in generating impromptu reports and getting answers to one-time, "what if" questions. For example, "what if there were a change in the styling of a product's design?" "How would material requirements and lead and delivery times be affected?" Access to an IMAGE data base can provide immediate answers to such questions. Special formatting features such as forms, titles, page and column headings, data sorting by categories, subtotals, totals and averages all contribute to readable, understandable reports.

HP 1000 FEATURES

A new sprint speed computer is the nucleus of each HP 1000 System. It provides the important performance necessary for multi-programming, multi-user application environments. The new processor executes most instructions 60% – 100% faster and performs floating point operations 250% faster than its predecessor.

Processor growth power is built into each HP 1000 in a number of ways. First, a variety of peripheral and software options enable users to upgrade from the smallest HP 1000 model to the largest as their requirements expand. Next, fast, low cost, semi-conductor, main memory capacity of up to 608k bytes is twice that of comparable systems. And the HP 1000's new processor uses a very high performance control processor, or "computer within a computer," that enables a user to increase computational horsepower almost at will. A simple terminal oriented language and editor can be used to create and load small routines, or even entire applications or operating systems into the control processor's large address space for faster execution. One HP implementation of this concept increases FORTRAN performance from 2 to 20 times over the non-microcoded execution. The HP 1000 is a computer that strenuously resists being obsoleted by demands for more processing speed.

Advanced Display. The 1000 System also takes advantage of the innovations in the fast new 9600 baud, high resolution screen HP 2645 Display Station. Pocket sized minicartridges can be used for convenient storage of programs close to your heart, or for real time logging of data entry transactions on the dual cartridges.

The 2645's "Soft Keys" can be programmed to automatically enter multiple keystroke sequences.

With a single stroke of a user defined "Soft Key," you can load or compile a program, query a data base, or monitor the status of multiple tasks. (Refer to figure 3.)

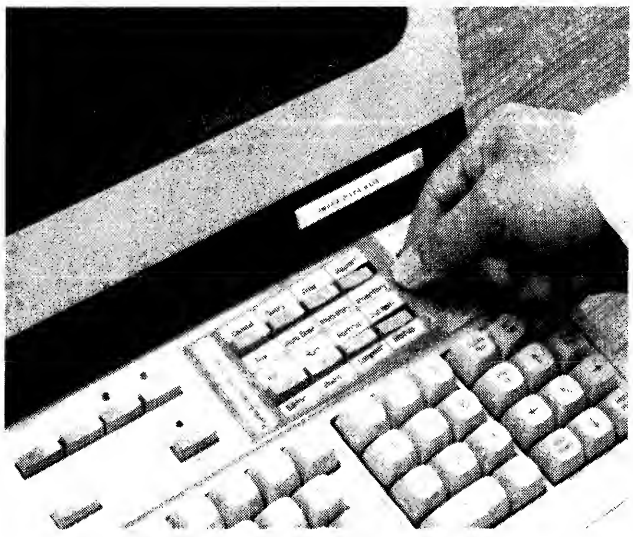


Figure 3. The 2645's Soft Keys

Multi-Talented Operating System. The HP 1000 System has a voracious appetite for many different applications because its power can be applied in many ways. The HP 1000 operating system orchestrates interactive program development from multiple terminals concurrently with batch processing. Multi-lingual programming – in FORTRAN, ALGOL, HP Assembly and Multi-User Real-Time BASIC – allows users to communicate with the system in the language that best suits their requirements.

Rapid Access Disc. The performance demands of disc active applications, such as data base management and multi-terminal applications, are met with latest track-follower disc technology and the micro-processor based control unit of the system's HP 7905 Disc. The HP 1000's disc storage capacity of up to 120m bytes allows the construction of a data base large enough to serve most small to medium-sized organizations.

Standard Instrument Connection. Thanks to the Hewlett-Packard Interface Bus (HP-IB*) the HP 1000 can be put to work in nearly any kind of automated electronic or electrical testing measurement and control applications. With an HP-IB interface kit, the HP 1000 can control multiple clusters of instruments, each consisting of up to 14 HP-IB compatible instruments. More than 100 HP-IB interfaceable measurement and test instruments are now available from Hewlett-Packard and other instrument manufacturers.

Network Connections. If you use more than one computer within your organization, linking these "islands of automation" can give you more complete control of your operations and significantly greater flexibility in collecting and managing information. Peripheral costs can be shared, data can be managed centraliy, and system availability improved through the use of networks.

*The Hewlett-Packard interface Bus (HP-IB) is HP's implementation of IEEE Standard 488 and identical ANSI Standard MC1.1, "Digital Interface for Programmable Instrumentation".

APPLICATIONS IN COMPUTATION

Either of the smaller 1000 Systems, Model 30 (desk-style) or 31 (in upright cabinets) would be an appropriate choice as the main computation resource of a medium-size engineering design facility. Like all 1000's, these are programmable in BASIC, FORTRAN, ALGOL, and HP Assembly languages. As many as four terminals can be active, developing separate new programs, while the system is also executing other programs. Especially in the lab, new micro-software, and dynamically-available new micro-control space will enhance computing operations. Typical jobs would include computer-aided design projects, artwork development, personal programs, and product simulation. The system could, at the same time, serve such administrative needs as critical-path project management, cost control, and analysis and development of standard costs, specifications, and bills of materials.

APPLICATIONS IN INSTRUMENTATION

With new speed, and available HP instrument-control hardware and software, the 1000 System offers advantages as controller of instrument clusters or automatic test stations.

The HP catalog now lists more than 40 standard products that are programmable via the HP Interface Bus (Hewlett-Packard's implementation of IEEE Standard 488). These include counters, digital multimeters, printers, scanners, network and spectrum analyzers, synthesizers and tape recorders. 22 other manufacturers now offer bus-programmable products. Standard plug-in interface cards will interconnect the 1000 System as controller of one or more clusters of off-the-shelf HP-IB instruments, forming automated systems of endless variety.

The 1000 System is also suitable as a controller of measurement and control systems such as the HP 9611R, in harsh industrial environments.

APPLICATIONS IN OPERATIONS MANAGEMENT

1000 Systems offer new solutions to operations-management problems that arise in many manufacturing organizations.

Used in a typical manufacturing department, to get the parts explosion under control and achieve efficient shop floor control, one of the larger 80 or 81 models, with standard IMAGE/1000 data-base management software, disc, tape, and printer can readily give the manufacturing administrator improved control over inventory, materials, planning, and production scheduling. With networking software it can also deliver edited, up-to-date reports to other data centers. With the speed of all 1000's, the larger capacity of Models 80 and 81 makes them attractive as central controller of a network of similar machines.

Any 1000 System will serve well in a data collection system, deriving information from multiple, dispersed data-entry devices for a manufacturing data base.

This well-balanced combination of HP 1000 computer system power, and IMAGE/1000 information-handling usefulness makes dynamic, user-oriented information handling a reality at an exceptionally low cost. An HP 1000 data base management system can be dedicated to a single department or to an entire company. Or several data base management systems can be linked in network, allowing each local data base to be shared by users throughout the network. Prices for complete data base management systems from Hewlett-Packard start at $61,200.

Four basic HP 1000 system models priced from $33,500 to $62,200, make the HP 1000 the lowest cost member of the Hewlett-Packard family of major computer systems. HP 1000, an ideal starter system for experienced OEM's and end users, is followed by the HP 2000 Timesharing and the powerful multi-programming, multi-lingual HP 3000 Series II.

The HP 1000 for Computation, Instrumentation, and Operations Management Applications

about the HP 2000

software tips

HP COMPUTER SYSTEMS EDITOR/2000

Dan Jorgenson
HP General Systems

EDITOR/2000, the new Text Editor and Formatter for the 2000 Computer System was introduced in September, 1976.

It can be used to prepare and manipulate textual material, typically contracts, proposals, technical manuals, specifications and legal documents. Special facilities are provided to edit the images of source language programs, such as BASIC programs, to be executed locally on an HP 2000 System, or COBOL programs, to be compiled and executed on a host computer via remote job entry. Source programs for other host programming languages (e.g., FORTRAN, PL1, etc.), also may be locally edited on the HP 2000 for remote job entry to host systems.

The new Editor provides the user with the following features:

- One command changes all occurrences of a word or character string to another word or string.
- Lines can be inserted, deleted, moved, copied, replaced, modified, renumbered, filled and justified.
- Extensive formatting options to set margins, print headings, adjust spacing, automate paging operations, and facilitate hyphenation of words.
- Automatic margin control carries text to the next line when it extends beyond a margin.
- Lines may be added or modified in any order desired.
- Files can be used to store EDITOR/2000 commands to be executed in unattended mode.
- Files can be created and developed from a terminal; text can be added, inserted, merged and replaced from a terminal or data file.
- Utilizes visual display features of HP 2640 series terminals.
- Permits document storage, retrieval and update using tape cartridges of the HP 2644 terminal.
- All editing is performed in an intermediate file and copied into a permanent TEXT file upon command.
- Special editing environments for BASIC and COBOL source programs.
- EXPLAIN command displays descriptions that define the use of each command.

EDITING TEXT – AN EXAMPLE

The simplicity and ease of using EDITOR/2000 is illustrated in the following example:

1. Typing-In The Rough Draft

   The rough draft may be typed into the system as fast as desired. The typist need not worry about the loss of data beyond margin settings, centering headings, numbering pages, etc.

   The following example illustrates the typical use of a few commands to edit and format text.

Section I

Introducing Editor/2000

1.1 What is Editor/2000

Editor/2000 can be used to prepare and manipulate textual material such as proposals, technical manuals, specifications, and legal documents. It also can be used to edit the images of BASIC and COBOL statements.

Editor/2000 is a HP2000 System interactive text editor which is used to create, edit, and format documents. Revisions to these documents can be made from standard typewriter keyboard terminals or video display terminals located anywhere near a telephone line and electrical power.

Figure 1. Rough draft, unedited

By typing the Command: SET AUTOMARGIN = ON, EDITOR/2000 will automatically fold lines of text that exceed the boundaries of the margin, onto subsequent lines. As text is typed, it is stored on the HP 2000 Computer System. It can be listed at any time on a terminal printer or line printer to produce a draft for revisions. EDITOR/2000 assigns reference line numbers at the left margin of the text, to identify portions of the text for the editing process.

```
 1   SECTION I
 2
 3   INTRODUCING EDITOR/2000
 4
 5   1.1 What is EDITOR/2000?
 6
 7   EDITOR/2000 can be used to prepare and manipulate
 8   textual material such as proposals, technical
 9   manuals, specifications, and legal documents. It
10   also can be used to edit the images of BASIC and
11   COBOL statements.
12
13   EDITOR/2000 is a HP 2000 System interactive text
14   editor which is used to create, edit, and format
15   documents. Revisions to these documents can be
16   made from standard typewriter keyboard terminals
17   or video display terminals located anywhere near a
18   telephone line and electrical power.
19
```

Figure 2. First typed copy

2. Revising the Text

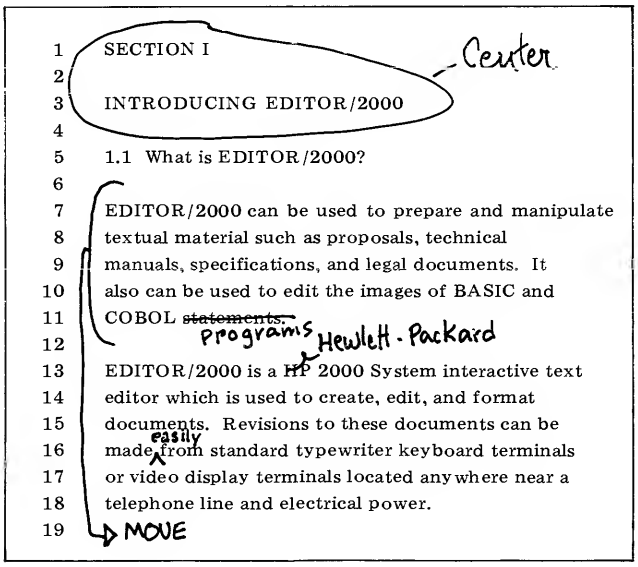


Figure 3. Edit marks on the first typed copy

The following commands are used to make the revisions to the document in figure 3. The typist's input has been shaded.

1. Change HP to Hewlett-Packard in line 13:

```
>change "HP" to "Hewlett-Packard" in 13
 13 EDITOR/2000 is a Hewlett-Packard 2000 System
    interactive text
```

2. Insert the word "easily" before the word "from" in line 16:

```
>insert 16("from"):
 16 made from standard typewriter keyboard terminals
    :easily
 16 made easily from standard typewriter keyboard
    terminals
```

3. Replace the contents of line 11:

```
>replace 11:
 11 COBOL statements.
 11 :COBOL programs.
```

4. Move the first paragraph:

```
>move 7:12 to 20
```

```
 1   SECTION I
 2
 3   INTRODUCING EDITOR/2000
 4
 5   1.1 What is EDITOR/2000?
 6
13   EDITOR/2000 is a Hewlett-Packard 2000 System
13   interactive text
14   editor which is used to create, edit, and format
15   documents. Revisions to these documents can be
16   made easily from standard typewriter keyboard terminals
17   or video display terminals located anywhere near a
18   telephone line and electrical power.
19
20   EDITOR/2000 can be used to prepare and manipulate
21   textual material such as proposals, technical
22   manuals, specifications, and legal documents. It
23   also can be used to edit the images of BASIC and
24   COBOL programs.
```

Figure 4. Revised copy

3. Formatting the Text

Revisions to the format of the text are made easily and quickly with the following commands. The typist's input has been shaded.

1. Set the right margin to 45 and enable hyphenation for the filling operation:

```
>set rbound=45,hyphenation=on
RBOUND =        45
HYPHENATION= ON
```

2. Fill each line with as many words as possible within the margins:

```
>fill 13:last
Hyphenate the following words:
cre ate
   -
vi deo
   -
manip ulate
      -
```

3. Justify the contents of the paragraphs:

```
>justify 13:last
```

4. Center the headings on lines one and three:

```
>center 1,3
```

4. **Producing the Final Copy**

By specifying a variety of format options in a LIST command, the typist can produce a final copy with titled and numbered pages, specific margin settings, and line spacing.

Issuing the LIST command:

```
>listq all,paginate,to printer
```

SECTION I

INTRODUCING EDITOR/2000

1.1 What is EDITOR/2000?

EDITOR/2000 is a Hewlett-Packard 2000 System interactive text editor which is used to create, edit, and format documents. Revisions to these documents can be made easily from standard typewriter keyboard terminals or video display terminals located anywhere near a telephone line and electrical power.

EDITOR/2000 can be used to prepare and manipulate textual material such as proposals, technical manuals, specifications, and legal documents. It also can be used to edit the images of BASIC and COBOL programs.

Figure 5. Final copy

By using the LIST command after the revisions have been made, the typist can immediately print an updated copy of the draft at the system line printer, a remote terminal printer, or CRT display.

TAKING ADVANTAGE OF HP 2640 SERIES TERMINALS

EDITOR/2000 takes advantage of the visual display features of the HP 2640 Series CRT terminals and mini cartridges of the HP 2644 terminal in the following ways:

1. Characters which exceed the right margin when the AUTOMARGIN feature is not being used, will appear on the screen with an inverse video (white) background. Characters appearing in inverse video will be discarded by EDITOR/2000.

2. Setting and clearing tabs through EDITOR/2000 will automatically set and clear the tabbing mechanism in the 2640.

3. When using the AUTOMARGIN feature, characters which exceed the right margin will automatically appear on the next line.

4. Inserted characters automatically appear in the displayed line.

5. Documents may be prepared off-line using mini tape cartridges in the HP 2644. On command, EDITOR/2000 will transfer the document from the tape cartridge to the WORK file and automatically place the text within special margins.

6. Edited documents can be easily stored on the mini tape cartridges and later restored to an EDITOR/2000 WORK file.

SYSTEM REQUIREMENTS

EDITOR/2000 can be used with any HP 2000 System configured with the HP 2000 Operating System (HP 22687A) Revision 1628 or higher.

DOCUMENTATION SUPPLIED

EDITOR/2000 Reference Manual (22701-90001)

ORDERING INFORMATION

HP 22701A	EDITOR/2000 source code supplied on 800 bpi, 9-track magnetic tape, includes the manual.
HP 22701-100	Same as above, except supplied on 1600 bpi, 9-track magnetic tape.

Please contact your local HP representative for ordering details and prices.

bulletins

NEW SYSTEM NAME

The current name of the system **HP 2000 Access System** will be changed to **HP 2000 Computer System.** This naming convention was effective with the announcement of new prices and configurations for the HP 2000 Computer System (model no. 19700B) in October 1976. The word **Access** has served its purpose well in describing the HP 2000, and has helped to project an exciting image of

"getting easy access to computing power". This image certainly applies to the new HP 2000 Computer System (19700B). Its new software capabilities further expand and refine this concept.

Other naming convention changes are listed below:

Language Processor:	old name – 2000/Access BASIC new name – 2000 BASIC
Operating System:	old name – 2000 Access Operating System new name – 2000 Operating System

NEW ORDERING PROCEDURES FOR THE COMPUTER CURRICULUM PROJECT

Chris Doerr
HP General Systems

HP's Computer-Based Educational Materials (formerly known as the Computer Curriculum Project) are no longer available from HP – they must now be ordered from:

Scientific Press
Stanford Barn
Palo Alto, CA 94304
(415)322-5221

This arrangement was instituted to greatly streamline ordering procedures for customers, and to speed up the turnaround time. Some of the titles include:

Conversational Statistics
Interactive Forecasting
Cases in Computer and Model-Assisted Marketing: Planning
Cases in Computer and Model-Assisted Marketing: Data Analysis
Air Pollution
Computer Graphics
INQUIR Reference Manual
COBOL 2000 Workbook

Also included are a series of mathematics enrichment supplements, physics booklets, and social science workbooks.

For a catalog of available books or to place an order, write or call Mr. Paul Kelly at the above address.

REVISIONS TO SEPTEMBER, 1976 COMMUNICATOR ARTICLE

Tom Aguirre
HP General Systems

Please note the following revision to the 2000 Software Update section in the September, 1976 Communicator. Programs CIS500 and CIS503 on page 447 should be deleted.

software updates

LATEST SOFTWARE REVISION TO THE 2000 OPERATING SYSTEM: REVISION 1638

Scott Guthrie
Dan Jorgenson
HP General Systems

The article describes new capabilities and problems corrected in revision 1638 of the 2000 Operating System (22678A) released in October, 1976. Also included in this article is a guide to taking system cold dumps.

NEW CAPABILITIES

9600 bps. Synchronous Communication

The synchronous data communication transfer rate used for RJE and 2000 system to 2000 system applications range upward to 9600 bps. It was formerly limited to 4800 bps. However, when using the new Hewlett-Packard serial link terminal (discussed in the next section) concurrently with RJE and 2000 to 2000 communications the transfer rate should be limited to 4800 bps.

Support for HP Serial Link Terminal

The Hewlett-Packard serial link terminal subsystem (HP 92900A) is now supported as a transaction-oriented input/output device attached to the Communications processor of the HP 2000 Computer System. This subsystem consists of an HP 40280A computer interface card, one HP 3070A Serial Link Terminal cabling, diagnostics, and user reference documentation. Additional terminals may be supported by this interface in a hardwired, multidrop fashion.

The HP 3070A terminal is a compact desk top terminal (see figure 1) with a 16 digit numeric display, 10 digit numeric keypad, special function keys, and 15 prompting lights. Up to 31 3070A terminals may be connected to an HP 2000 Computer System through a single shielded, twisted pair cable up to 4 kilometers long (2.5 miles). Additionally up to 32 interactive terminals may be attached through the HP 2000's two 16 channel multiplexors. The capability of the 3070A can be extended by the addition of any of our one-hundred HP-IB* compatible measurement instruments and peripherals currently available.

*The HP-IB is Hewlett-Packard's implementation of the IEEE STD 488-1975 "Digital interface for programmable instrumentation".

Figure 1. HP 3070A Terminal

The software driver for the 3070A is included in revision 1638 and may be configured into the communication processor during system generation. The terminal is treated as an ASCII file device and controlled through BASIC language programs.

Applications for this terminal in manufacturing companies include time-keeping, inventory control, materials movement, production testing, and order processing. Many functionally similar applications exist within non-manufacturing organizations such as insurance companies, banks, and hospitals.

Please contact your HP representative for further information on the HP 3070A Serial Link Terminal.

PROBLEMS CORRECTED IN REVISION 1638

I/O Processor

1. Repeated power failures could cause the I/O processor to crash.
2. Possible port lockouts after an ANNOUNCE, SLEEP, or HIBERNATE Commands.
3. Data Communications/RJE Facility:
   a. The IOP would crash if multiple print streams were retrieved from an RJE JOB LISTER (JL)
   b. The 11-7-8 punch was translated incorrectly during CDC RJE.
   c. "Data Check" errors caused by internal timing constraints while operating 9600 bps communications links have been eliminated in almost all circumstances. Systems operating HP 3070 link terminal subsystems will experience unacceptable error rates with RJE communications above 4800 bps.
   d. During 2000 to 2000 communications, CTL's to a remote lineprinter were not handled correctly.
   e. Systems with an HP 2894 card reader/punch could experience erratic operations.
4. Lower case characters are shifted to upper case instead of blanks for printing on the HP 2767 line printer.
5. Operation of dial-up ports on a 16 port system was unpredictable.
6. Parity bits were not ignored ("stripped") on paper tapes that were punched or read in the "No Parity" mode.
7. Several characters were translated incorrectly on Call/360 terminals.
8. The IOP could crash in very heavily loaded systems when using RJE and/or multiple ASCII File devices.

System Processor

1. Mat read statement did not read array elements from strings in ASCII files correctly.
2. Arguments for the PAUSE statement must be less than 32,768. No error was indicated for arguments in the range of 32,768 to 65,535. The manual specifies incorrect ranges (Manual not corrected in this release).

PROBLEMS KNOWN BUT NOT YET CORRECTED

1. The processors in the system are susceptible to static electricity. This is a common reason that systems become unresponsive (i.e., "Ports Lock Up"). A hardware revision is available to reduce this problem. All normal procedures to minimize static electricity around the system are strongly recommended.
2. The system does not support print positions 133 through 136 on the HP 2613, HP 2617 and HP 2618 lineprinters. (To be corrected in the next release).
3. Often, the system does not report to the user or to the console that a lineprinter went "off-line" for any reason while printing. (To be corrected in the next release).
4. RJE output from CDC hosts does not support print positions 133 through 136. (To be corrected in the next release).
5. Dial up ports are enabled inappropriately if a power failure occurs while the system is in the SLEEP state. (To be corrected in the next release).

DOCUMENTATION

New updates to the Basic Reference and Operator's Manual based on the features of revision 1638 have been printed. These updates can be obtained by specifying the following information:

Manual Update #2 (including #1), dated 9/76 to HP 2000 Basic Reference Manual, Second Edition, 5/76, Manual Number 22687-90001.

Manual Update #2 (includes #1), dated 9/76 to HP 2000 Operator's Manual, Second Edition 5/76, Manual Number 22687-90005.

OBTAINING REVISION 1638

Customers under warranty or service contract will receive Revision 1638 at no cost including one copy each of the manual updates. Customers not under warranty or service contract that wish to order Revision 1638 should contact their local HP representative for ordering details.

HP 2000 COLD DUMP PROCEDURES

In the past, there has been some misunderstanding of the system problem reporting procedure involving cold dumps. It is the purpose of this section to clarify this procedure. Good cold dumps are essential in diagnosing software problems.

The Cold Dump Routine

The cold dump routine is a memory-resident program designed to assist in preserving information necessary to identify the cause of software failures. Correct usage minimizes system down-time because the routine captures essential data for subsequent off-line analysis. This feature allows recovery procedures to proceed in parallel with the fault diagnosis. Note that the cold dump routine generally is not useful for diagnosing hardware problems.

If software failures occur, it is important to record as much information as possible about the state of the system before the failure. A special report form provides space for data pertinent to the cold dump procedure; for example, register contents, time of failure, and anomalous circumstances preceding the failure. A sample Cold Dump Report is included in Appendix H of the HP 2000/Access Operator's Manual. Always fill out the report completely as part of the cold dump procedure.

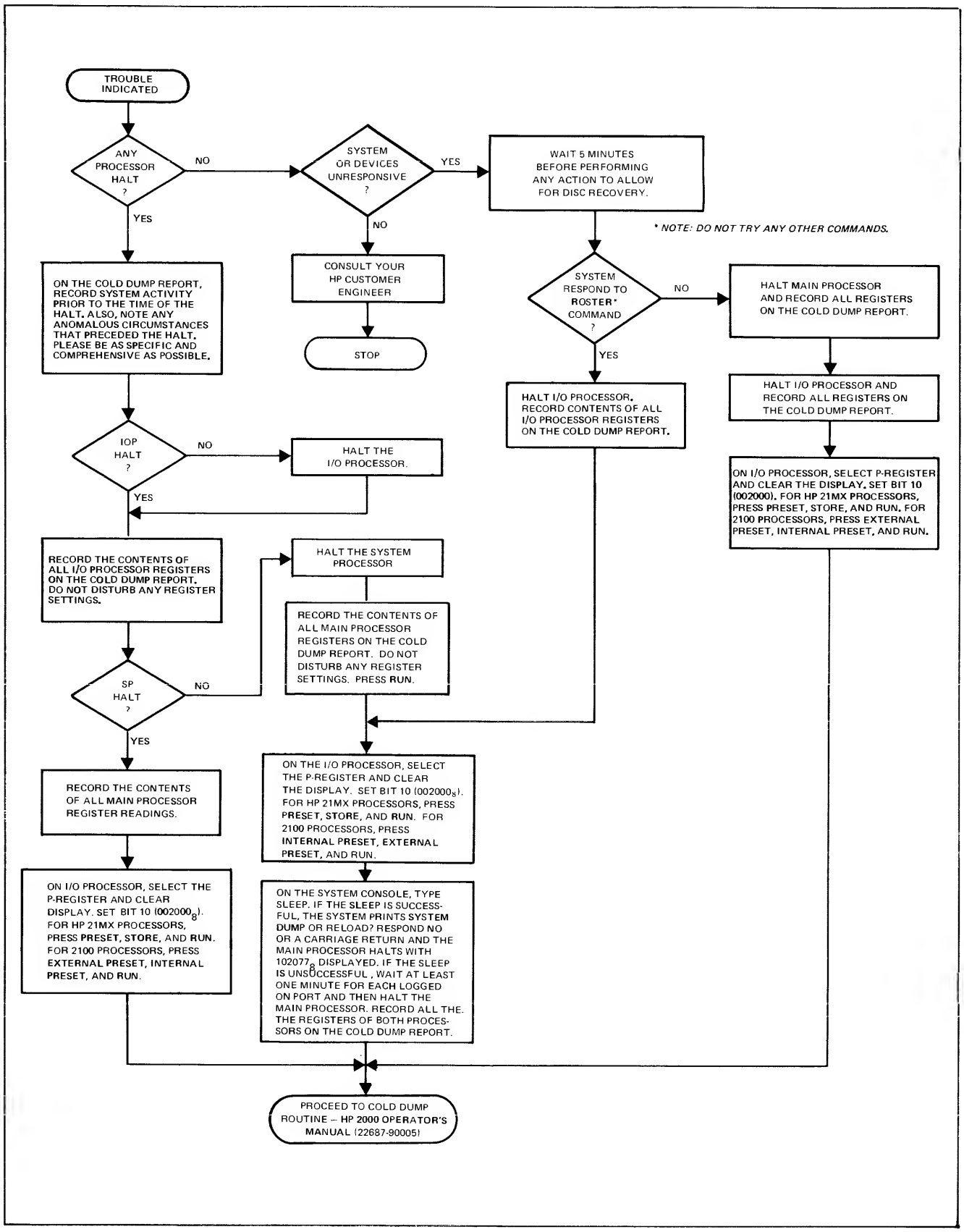
If the system fails, halt codes may be displayed in the display register of the processors. In Appendix B, tables B-1 and B-2 list the halt codes, their meanings, and possible corrective action. When these halts require using the cold dump routine, follow the flowchart attached to set the system up for a successful dump.

NOTE

A cold dump is not useful if either the main processor or I/O processor parity light is ON (indicating a parity error occurred). If this happens, contact your HP customer engineer and then consult section 5 or System Recovery Procedures in the Operator's Manual.

If a problem with normal system operation is encountered, perform the following procedure:

(See page 536 for Logic Diagram)


TROUBLE INDICATED
ANY PROCESSOR HALT ?
NO
YES
SYSTEM OR DEVICES UNRESPONSIVE ?
YES
NO
WAIT 5 MINUTES BEFORE PERFORMING ANY ACTION TO ALLOW FOR DISC RECOVERY.
* NOTE: DO NOT TRY ANY OTHER COMMANDS.
CONSULT YOUR HP CUSTOMER ENGINEER
STOP
ON THE COLD DUMP REPORT, RECORD SYSTEM ACTIVITY PRIOR TO THE TIME OF THE HALT. ALSO, NOTE ANY ANOMALOUS CIRCUMSTANCES THAT PRECEDED THE HALT. PLEASE BE AS SPECIFIC AND COMPREHENSIVE AS POSSIBLE.
SYSTEM RESPOND TO ROSTER* COMMAND ?
NO
YES
HALT MAIN PROCESSOR AND RECORD ALL REGISTERS ON THE COLD DUMP REPORT.
HALT I/O PROCESSOR. RECORD CONTENTS OF ALL I/O PROCESSOR REGISTERS ON THE COLD DUMP REPORT.
HALT I/O PROCESSOR AND RECORD ALL REGISTERS ON THE COLD DUMP REPORT.
ON I/O PROCESSOR, SELECT P-REGISTER AND CLEAR THE DISPLAY. SET BIT 10 (002000). FOR HP 21MX PROCESSORS, PRESS PRESET, STORE, AND RUN. FOR 2100 PROCESSORS, PRESS EXTERNAL PRESET, INTERNAL PRESET, AND RUN.
IOP HALT ?
NO
YES
HALT THE I/O PROCESSOR.
RECORD THE CONTENTS OF ALL I/O PROCESSOR REGISTERS ON THE COLD DUMP REPORT. DO NOT DISTURB ANY REGISTER SETTINGS.
HALT THE SYSTEM PROCESSOR
RECORD THE CONTENTS OF ALL MAIN PROCESSOR REGISTERS ON THE COLD DUMP REPORT. DO NOT DISTURB ANY REGISTER SETTINGS. PRESS RUN.
SP HALT ?
NO
YES
RECORD THE CONTENTS OF ALL MAIN PROCESSOR REGISTER READINGS.
ON I/O PROCESSOR, SELECT THE P-REGISTER AND CLEAR THE DISPLAY. SET BIT 10 (002000_8). FOR HP 21MX PROCESSORS, PRESS PRESET, STORE, AND RUN. FOR 2100 PROCESSORS, PRESS INTERNAL PRESET, EXTERNAL PRESET, AND RUN.
ON I/O PROCESSOR, SELECT THE P-REGISTER AND CLEAR THE DISPLAY. SET BIT 10 (002000_8). FOR HP 21MX PROCESSORS, PRESS PRESET, STORE, AND RUN. FOR 2100 PROCESSORS, PRESS EXTERNAL PRESET, INTERNAL PRESET, AND RUN.
ON THE SYSTEM CONSOLE, TYPE SLEEP. IF THE SLEEP IS SUCCESSFUL, THE SYSTEM PRINTS SYSTEM DUMP OR RELOAD? RESPOND NO OR A CARRIAGE RETURN AND THE MAIN PROCESSOR HALTS WITH 102077_8 DISPLAYED. IF THE SLEEP IS UNSUCCESSFUL, WAIT AT LEAST ONE MINUTE FOR EACH LOGGED ON PORT AND THEN HALT THE MAIN PROCESSOR. RECORD ALL THE REGISTERS OF BOTH PROCESSORS ON THE COLD DUMP REPORT.
PROCEED TO COLD DUMP ROUTINE – HP 2000 OPERATOR'S MANUAL (22687-90005)

documentation

The following tables list currently available customer manuals for HP 2000 Systems products. This list supersedes the list in the last issue of the **Communicator.**

The most recent changes to the tables are indicated for easy reference. Prices are subject to change without notice.

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package complete the Update Order Form in the back of the **Communicator** and mail the form to:

Software/Publications Distribution
5303 Stevens Creek Blvd.
Santa Clara, CA. 95050

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manual(s) you need on the Corporate Parts Center form supplied at the back of the **Communicator.**

A few words about documentation terms:

New — A new manual refers only to the first printing of a manual. When first printed, a manual is assigned a part number.

Revised — A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.

Update — An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number; it is automatically included when you order the manual with which it is associated.

PART NUMBER	2000 E	2000 F	HP 2000	MANUAL TITLE	PRICE†	PUBLICATION DATE	CURRENT UPDATE
02000-90055		X		2000C/2000F IDF Author's Manual	$ 8.50	1/73	8/74
02000-90080		X		2000E to 2000F Conversion Guide	1.00	4/76	
19665-90001			X	2000/F to 2000 Computer System Upgrade Kit and Conversion Program Manual	2.00	8/76	
19665-90002			X	2000/F to 2000 Computer System Educational Application Upgrades	1.00	2/76	
22687-90001			X	2000 BASIC Reference Manual	10.00	5/76	9/76
22687-90005			X	2000 Operator's Manual	10.00	5/76	9/76
22687-90007			X	2000 System Operator's Pocket Guide	1.50	5/76	
22687-90003			X	BASIC Pocket Guide	1.50	10/76 *R	
02000-90048	X			BASIC/2000 Level E Reference Manual, Timeshared	10.00	9/75	
02000-90049	X			BASIC/2000 Level E System Operator's Manual, Timeshared	5.00	9/74	8/75
5952-4490	X			BASIC/2000 Level E Pocket Guide, Timeshared	0.15	10/74	
02000-90073		X		BASIC/2000 Level F Reference Manual, Timeshared	7.50	12/75	
02000-90074		X		BASIC/2000 Level F System Operator's Manual, Timeshared	10.00	8/76	
5952-4491		X		BASIC/2000 Level F Pocket Guide, Timeshared	0.15	8/75	
24387-90001		X		Basic Analysis and Mapping Program Manual	18.00	6/74	5/75
24387-90002		X		Basic Analysis and Mapping Program Pocket Guide	1.00	6/74	
24384-90001		X	X	College Information System — System Overview	5.00	6/74	9/76
24384-90003		X	X	College Information System Reference Manual	19.00	9/75	9/76
24384-90005		X	X	College Information System — Technical Manual	95.00	5/75	
24383-90001		X		Course Writing Facility Reference Manual	15.00	5/74	
22692-90001			X	Course Writing Facility Reference Manual	16.50	12/75	

PART NUMBER	2000 E	2000 F	HP 2000	MANUAL TITLE	PRICE†	PUBLICATION DATE	CURRENT UPDATE
5951-1381		X		DOS-M/2000C Timeshared BASIC File Handler	1.00	5/71	
22701-90001			X	EDITOR/2000 Reference Manual	7.00	9/76	
20352-90001		X		Educational Budget and Accounting System – System Overview	10.00	6/74	
20352-90002		X		Educational Budget and Accounting System – Reference Manual	10.00	3/75	4/76
20352-90003		X		Educational Budget and Accounting System – Technical Manual	75.00	3/75	
20353-90001		X		Educational Payroll System – System Overview	3.50	10/74	
22700-90001			X	FCOPY/2000 Reference Manual	4.50	1/76	
22693-90003			X	HP MATH Curriculum Guide	17.50	7/75	
22693-90002			X	HP MATH Proctor's Manual	6.50	7/75	
22693-90001			X	HP MATH Teacher's Handbook	5.50	7/75	
20310-90007		X		HP MATH Curriculum Guide	20.00	7/74	
20310-90005		X		HP MATH Proctor's Manual	5.00	9/74	
20310-90001		X		HP MATH Teacher's Handbook	5.00	9/75	
22691-90003			X	Instructional Dialogue Facility Author's Manual	13.00	9/75	
22691-90004			X	Instructional Dialogue Facility Author's Pocket Guide	3.00	9/75	
22691-90002			X	Instructional Dialogue Facility Course Developer's	5.00	9/75	
22691-90001			X	Instructional Dialogue Facility Proctor's Manual	6.00	9/75	
20309-90005		X		Instructional Dialogue Facility Author's Pocket Guide	3.50	10/74	
20309-90003		X		Instructional Dialogue Facility Course Developer's Manual	6.00	8/74	
20309-90001		X		Instructional Dialogue Facility Proctor's Manual	10.00	9/74	
22690-90001			X	Instructional Management Facility Proctor's Manual	6.50	9/75	
22690-90002			X	Instructional Management Facility System Manager's Reference Manual	4.50	9/75	
20308-90001			X	Instructional Management Facility Proctor's Manual	7.00	9/74	
20308-90003		X		Instructional Management Facility System Manager's Manual	5.00	10/74	
22687-90009			X	Learning Timeshare BASIC	3.50	5/76	
20243-90001			X	Source Data Entry/2000 Reference Manual	5.00	2/76	
20240-90001			X	Telecommunications Supervisory Package/2000 Manager's Manual	5.00	1/76	
20240-90002			X	Telecommunications Supervisory Package/2000 User's Manual	3.50	1/76	
20311-90001		X		Timeshared Graphics for Tektronix Terminals	7.00	8/74	
20311-90003		X		Timeshared Graphics Plotting Package	5.00	8/74	

*N = New Manual (Refer to the Bulletin section)

*R = Revised Manual

† Prices listed are subject to change without notice.

training schedule

The schedule for customer training courses on General Systems Division Products is outlined below and in the HP 3000 section of this publication. Included here are 2000 courses for the 3 month period, November, 1976 through January, 1977.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE

Course Dates and Training Center Location

COURSE NUMBER	COURSE TITLE	LENGTH	PRICE	WESTERN TECHNICAL CENTER – SANTA CLARA	EASTERN TECHNICAL CENTER – ROCKVILLE
22973A	2000 User	5 days	$500	1/24/77 3/7/77	Scheduled upon sufficient demand.

Registration

Requests for enrollment in any of the above courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time your Sales Office requests a registration.

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than one week before the class begins.

EASTERN TECHNICAL CENTER	WESTERN TECHNICAL CENTER
Hewlett-Packard 4 Choke Cherry Road Rockville, Maryland 20850	Hewlett-Packard 5303 Stevens Creek Blvd. Santa Clara, Calif. 95050

about the HP 3000

software tips

COBOL VERSION C RELEASED (32213C)

John Page
General Systems

For HP System users who want to offload some COBOL work from a big mainframe to a local 3000, one more barrier has been knocked down. Now with COBOL Version C customers with large 370 programs, for example, will now find their conversion task eased by an order of magnitude.

COBOL/3000 Version C differs from Version B in the way the compiler's internal symbol table is handled. In the old version, this table would overflow on programs more than about 3000 lines in length (or even less if the program had a large number of symbols), causing a compile-time abort which was very difficult to get around. In fact, the only viable solution was to break up the program into sub-programs and compile them separately – a very painful job.

Version C has a symbol table which extends itself onto disc when it gets too large, *effectively eliminating program size restrictions.*

COMPATIBILITY

Apart from the change described above, there are no differences between Version B & C. It will run on both Series I (CX) and Series II.

PERFORMANCE

Due to the fact that some extra code had to be added to the compiler, it now takes up slightly more space in memory. On the Series II this has caused the compiler to take about 10% longer to compile a program than the B Version.

On the Series I it will take between 40 and 100% longer. *For this reason the standard COBOL for a CX will remain Verion B.* However, in order to provide Series I customers with the capability to compile the occasional large program, they will be provided with the Version C in addition to B compilers, but it will be accessible through the :RUN command. In summary:

	VERSION B	VERSION C
Series I	:COBOL	:RUN COBOLC. . .*
Series II	Not Needed	:COBOL

MAINTENANCE

Both Version B & C will be maintained until further notice (Version A has already been obsoleted).

Run COBOLC. PUB.SYS COMMAND

An alternative way to call the COBOL compiler is by using the :RUN command. Before using the :RUN command, you must use file equations for the files normally specified on the :COBOL command. The format file designators are:

COBTEXT	*(textfile)*
COBLIST	*(listfile)*
COBUSL	*(uslfile)*
COBMAST	*(masterfile)*
COBNEW	*(newfile)*

Thus, to compile from the file MYSOURCE and send the listing to the line printer, you would use

```
:FILE COBTEXT=MYSOURCE
:FILE COBLIST;DEV=LP
```

before using the :RUN command.

Additionally, you must specify a PARAM=*parameternum* parameter on the :RUN command to indicate which files are present unless the default values are used. The value is between 0 and 31 as shown on the following table. Basically, the low order five bits in *parameternum* represent the five files which can be specified as shown below:

11	12	13	14	15
newfile	masterfile	uslfile	listfile	textfile

For example, to invoke the compiler with the *textfile* and *listfile* present, you would use the command:

```
:RUN COBOLC.PUB.SYS;PARM=3
```

PARM Values

PARAMETERNUM	FILES PRESENT
0	None
1	*textfile*
2	*listfile*
3	*listfile, textfile*
4	*uslfile*
5	*uslfile, textfile*
6	*uslfile, listfile*
7	*uslfile, listfile, textfile*
8	*masterfile*

9	*masterfile, textfile*
10	*masterfile, listfile*
11	*masterfile, listfile, textfile*
12	*masterfile, uslfile*
13	*masterfile, uslfile, textfile*
14	*masterfile, uslfile, listfile*
15	*masterfile, uslfile, listfile, textfile*
16	*newfile*
17	*newfile, textfile*
18	*newfile, listfile*
19	*newfile, listfile, textfile*
20	*newfile, uslfile*
21	*newfile, uslfile, textfile*
22	*newfile, uslfile, listfile*
23	*newfile, uslfile, listfile, textfile*
24	*newfile, masterfile*
25	*newfile, masterfile, textfile*
26	*newfile, masterfile, listfile*
27	*newfile, masterfile, listfile, textfile*
28	*newfile, masterfile, uslfile*
29	*newfile, masterfile, uslfile, textfile*
30	*newfile, masterfile, uslfile, listfile*
31	*newfile, masterfile, uslfile, listfile, textfile*

3000 SERIES II TERMINAL TIPS

Rita Williams
HP General Systems

Below is a matrix of terminal types, representative terminals, and delay characters at various speeds. This matrix is provided to aid users who must use terminals other than the ones listed below. With this information, such users can select the termtype most *likely* to succeed!

FOR HARDWIRED TERMINALS

Given: the system's default terminal is type 10, user wants to log on a terminal other than type 10 using TERM = parameter of :HELLO. If terminal doesn't respond to initial carriage return, control-F followed by carriage return should get the system to respond with the prompt.

FOR TERMINALS ON MODEMS

The Series II support group suggests configuring these terminals as term type 0. This will allow any supported terminal type to enter HELLO with TERM – parm – without using control-F as above.

Termtype	Representative Terminal	Delay Characters (LF, CR)** Speed (Characters per Second) 10	15	30	60	120	240	Character Echoed on Control-H
0	ASR 33	0,1						\
1	ASR 37		0,2					LF
2	ASR 35	0,1						\
3	Execuport	0,1	0,3	0,5				LF
4	Datapoint	0,0	4,0	4,0	4,0	4,0	4,0	EM (control-Y)
5	Memorex 1240	1,10	1,10	1,25	3,45			LF
6	Terminet	3,0	5,0	10,0		35,0		LF
9	MiniBee	0,0	4,0	4,0	4,0	4,0	4,0	null
*10	HP2640/44	0,0	0,0	0,0	0,0	0,0	0,0	null
*11	HP2640/44 full "enter" service	0,0	0,0	0,0	0,0	0,0	0,0	null
	undefined	0,1	4,0	4,0	4,0	4,0	4,0	\

**See appropriate terminal manual to determine output delays required when moving cursor.*
***Line Feed, Carriage Return*

HOW TO BUILD AN INTERFACE BETWEEN COBOL AND ANY MPE INTRINSIC FROM SPL

Terry Von Gease
HP General Systems

There are few (if any) MPE Intrinsics callable directly from COBOL. This is due to the way COBOL sends parameters (in the USING clause) and the way the various MPE Intrinsics expect to receive them.

Some MPE Intrinsics are "Option Variable". This means that some of the parameters are optional. When an option variable procedure is called one or more extra words will be passed along with the parameters to indicate which parameters are present. The extra word, or words, are passed, one word for each 16 possible parameters. Bits are set on or off to indicate which parameters are present and which are not. The bits in the extra word, or words, correspond to the actual parameter list on a one for one basis from right to left. When calling an option variable intrinsic it is the responsibility of the calling process to provide this additional information. SPL handles this automatically, however, COBOL will not provide the proper data so any option variable intrinsic cannot be called directly from COBOL.

Also many of the MPE Intrinsics are "Typed Procedures". A Typed Procedure returns a value (either one, two, or four words) that is independent of any passed parameters. This is a value generated by the procedure itself. For example, assume that we have an SPL program that is going to call the MPE Intrinsic BINARY. This intrinsic will convert a byte array containing ASCII numeric characters for a specified length to a one word integer and return this integer as the value of the intrinsic. For this example assume that we have a byte array called BUFFER that contains the character string "1234", we have a one word integer called LENGTH that contains the number 4 (4 characters in BUFFER), and we have another one word integer called RESULT which will be used to contain the result. The actual call to BINARY would look like this . . . . . .

```
RESULT:=BINARY(BUFFER,LENGTH);
```

Note that the ":=" is SPL for "replaced by", a single "=" means something else.

In this case the value of the integer RESULT is replaced by the binary numeric value of the character string BUFFER. COBOL has no way of coping with this returned value so this MPE Intrinsic is useless to us on that basis alone. (Not to mention what it does to the stack internally).

Another feature of many MPE Intrinsics is the use of the Condition Code to indicate the result of a call to a particular intrinsic. In the BINARY example above the condition code would be set to < (less than) if the character string contained one or more non-numeric characters (except a leading minus in this case), a > (greater than) if the result was beyond the capacity of a one word integer (less than −32768 or greater than +32767), or = (equal) if the call functioned properly and the returned value in RESULT is valid. In SPL it is a simple matter to interrogate the condition code after a call, we simply say . . . . . .

```
IF < THEN . . .     (if the condition code is less than)
IF > THEN . . .     (if the condition code is greater than)
IF = THEN . . .     (if the condition code is equal)
IF < > THEN . . .   (if the condition code is not equal)
IF < = THEN . . .   (if the condition code is less than or equal)
IF >= THEN . . .    (if the condition code is greater than or equal)
```

What follows the THEN statement in each case is code for whatever you want to do if the statement is true.

Since COBOL has no way of interrogating the condition code we would have no way of knowing whether or not the call to BINARY worked properly even if we could call BINARY directly.

Furthermore, if we examine the description of the BINARY intrinsic in the *MPE Intrinsics Manual* we find that not only is BINARY a procedure that sets the condition code but the character string parameter must be a byte array passed by reference (COBOL passes word arrays by reference, so that's out) and the length parameter is a one word integer (so far so good) passed by value (so we lose here also).

The difference between a reference and a value parameter is this: When a reference parameter is used the ADDRESS of the parameter is passed; when a value parameter is used the actual value of the parameter is passed. In the case of our call to BINARY, the address of the byte array BUFFER and the binary value 4 are passed. Let's take a look and see what actually happened when we called BINARY . . . . . .

We know (or we should know) that we have something called a stack for our program. The stack contains all of the data area for our program and is referenced by the STACK or S register (there are other references but S is all we need to consider for this example).

When the SPL compiler encounters the call to BINARY it generates code which does the following:

1. SPL interrogates the MPE Intrinsic file to determine the various characteristics of BINARY.

2. SPL notices that this procedure will return a one word integer value. To accommodate this a zero is loaded onto the stack and the S register is incremented by 1.

3. SPL sees that the first parameter is by reference so

it places a one word address of the first variable specified (BUFFER in this case) on the stack and the S register is incremented by 1.

4. SPL sees that the next parameter is passed by value so it places the value of the specified parameter (LENGTH in this case) on the stack and the S register is incremented by 1.

5. SPL sees that all of the parameters have been taken care of so it generates a PCAL machine instruction for BINARY.

6. PCAL places 4 more words on the stack and increments the S register by 4. These 4 words are referred to as a "Stack Marker" and contain all of the information needed to return to the proper place in the calling program.

7. PCAL does whatever is necessary to make BINARY present in the system and transfers control to it.

8. BINARY does whatever it is that BINARY's do. One of the things that it will do for sure is place the result value back down in the stack in the location of the first word that SPL placed there.

9. BINARY exits. When this happens all of the passed parameters (both of them in this case) and the 4 word stack marker are removed from the stack and the S register is decremented by the proper value (6 in this case). Note that the first location that was for the returned value is NOT removed from the stack. The condition code is set and control returns to the proper place in the calling program.

10. SPL remembers that this procedure will return a one word value so it removes it from the stack and places it in RESULT. The S register is now decremented by 1.

11. We are now back at the start of the next sequential instruction from where we left and the stack is back to what it was before the call to BINARY. The pot's right.

All of this may or may not make any sense at this time but it does serve to point out what we must do to get at any MPE Intrinsic from COBOL. We must construct an interface to resolve the parameter types, return variables, and the condition code. At this time a few words about COBOL data types may well be in order . . . . .

1. Any field included in a USING clause **MUST** be located on a word boundary. This is easily accomplished by defining everything that may be used as a parameter at an 01 or 77 level. If this proves annoying then word alignment may be insured by using the SYNC clause or merely by insuring that there is ALWAYS an EVEN number of bytes between parameters (this method is not recommended).

2. COBOL keeps numeric data in four forms:

   A. The display numeric form of PIC S9(3). All this is a string of ASCII numeric characters.

   B. The edited numeric form of PIC ZZZZZ. This is still a string of ASCII characters.

   C. The COMPUTATIONAL-3 form of PIC S9(3) COMP-3. This, of course, is packed decimal information with 2 digits per byte except for the last byte which contains 1 digit and the sign.

   D. The COMPUTATIONAL form of PIC S9(3) COMP. This is numeric data in integer form. COBOL uses no floating point data, all binary numeric data is kept in one to three word integers. 4 or less digits are one word, from 4 to 9 digits are two words, and greater than 9 digits are four words.

   Generally only numerical data of type COMPUTATIONAL will be of any use to us for passing numbers. And of the COMPUTATIONAL types only the one and two word integers will be used.

Now let us define and construct an interface so COBOL may call BINARY. The simplest way to do this is to write the calling program in COBOL and then write the interface in SPL.

The portions of the COBOL program doing the calling should look something like this . . .

In the working-storage section . . . .

```
01 BUFFER PIC X(4).
01 LENGTH PIC S9(4) COMP.
01 RESULT PIC S9(4) COMP.
01 COND-CODE PIC S9(4) COMP.
```

In the procedure division . . . . (assume that the COBOL program has moved "1234" to BUFFER and 4 to LENGTH) . . . . .

```
CALL "CBINARY" USING BUFFER, LENGTH,
    COND-CODE, RESULT.
```

Note that we have called our procedure CBINARY and we have included a couple of extra parameters, as you surely must have guessed, to contain the condition code and the result value.

The SPL interface will have to look something like this . . . .

```
1.   $CONTROL SUBPROGRAM
2.   BEGIN
3.   PROCEDURE CBINARY(BUFFER,LENGTH,
        COND' CODE,RESULT);
4.   ARRAY BUFFER;
5.   INTEGER LENGTH,COND'CODE,RESULT;
6.   BEGIN
7.      BYTE ARRAY BBUFFER(*)=BUFFER;
8.      INTRINSIC BINARY;
9.      RESULT:=BINARY(BBUFFER,LENGTH);
10.     IF < THEN COND'CODE:=-1 ELSE IF > THEN
           COND'CODE:=1 ELSE COND'CODE:=0;
11.  END;
12.  END.
```

Note that the line numbers are NOT used for SPL but only included so we may reference each line for discussion . . .

Line 1: This puts the compiler into SUBPROGRAM mode.

Line 2: All SPL programs of any sort must start with a BEGIN statement.

Line 3: This declares the procedure and all of its parameters to the compiler.

Line 4: Each of the parameters included in the procedure declaration (line 3) must now be described. Since COBOL always passes word pointers by reference we declare BUFFER to a word array.

Line 5: The rest of the parameters are declared as integers since we have described them to COBOL as one word integers. Note that we had to change the name of COND-CODE slightly since SPL does not allow the use of hyphen. In fact the only special character SPL allows in names is the apostrophe.

Line 6: Like the program itself, each procedure must start with a BEGIN statement.

Line 7: This is a local data declaration and is the heart of the whole interface. We have declared a byte array that shares the same storage location as the word array BUFFER. We call this "Equivalencing" the two arrays. Byte arrays may always be equivalenced to word arrays but never word arrays to byte arrays since there is no certainty of having a byte array begin on a word boundary. We could gain or lose a leading byte.

Line 8: This declares that BINARY can be found in the MPE Intrinsic file.

Line 9: This is the actual call to BINARY. Note that we used the parameters LENGTH and RESULT just as they were passed but we had to use the equivalenced array BBUFFER for the string parameter. When SPL encounters the LENGTH parameter, even though this parameter was passed by reference, it will take the data found at the address that was passed. In other words, SPL will always resolve value parameters whether the value parameter is a literal or a variable. If, due to some oraculation, we knew that there were exactly four characters contained in BUFFER (or BBUFFER since both of these are really two references to the same thing) we could have called BINARY thusly . . . . . .

```
RESULT:=BINARY(BBUFFER,4);
```

In this case the SPL compiler would have immediately placed the binary value four on the stack rather than have gone to the storage location LENGTH and used what it found there. The results would have been identical in either case.

Line 10: This takes care of the condition code. The passed parameter COND-CODE is set to -1 of the condition code was <, +1 if the condition code was >, or 0 if the condition code was =.

Line 11: This terminates the procedure.

Line 12: This terminates the program.

Now that we have written the source for both the COBOL main program and the SPL subprogram we can compile all of this into one USL file and PREP it into a runnable PROGRAM file. To do this always make sure that all of the non-COBOL source is compiled first, then compile the COBOL source. This is because of the way COBOL must link things up internally.

What we have done is resolve any required byte arrays by declaring a local byte array that is equivalenced to the passed word array, included an extra parameter from COBOL to handle the return value, and included an extra parameter to represent the condition code. If a particular System intrinsic does not return a value then that parameter can be omitted. Likewise, if the condition code does not matter to you then there is no need to include a parameter for it either.

In any case the outline for the SPL interface must look like . . .

```
$CONTROL SUBPROGRAM
BEGIN
PROCEDURE name your procedure and state its parameter
   list in parentheses;
ARRAY declare any arrays for future equivalencing here;
INTEGER declare any single word integers here;
DOUBLE declare any integers with 4 to 9 digits here;
BEGIN
   BYTE ARRAY name of local byte array (*)=name of
      passed word array;
   INTRINSIC names of the System Intrinsics that you
      want to call;
   call the System Intrinsic;
   set up the condition code;
END;
END.
```

This all looks quite complicated but be assured after the first time you'll love it.

We have not covered all of the possibilities here. We have only tried, through a representative sample, to show what generally must take place for an interface to function properly. Perhaps one important factor that was not brought out is that any number of interface procedures can be contained in one SPL source file. Merely stick them in between the main BEGIN/END pair like this . . .

```
$CONTROL SUBPROGRAM
BEGIN

PROCEDURE first procedure;
BEGIN
code for first procedure;
END;

PROCEDURE next procedure;
BEGIN
code for this procedure;
END;

and so on . . .

END.
```

And there you are.

JOURNEY FROM AN ABORT MESSAGE TO A LINE OF SOURCE CODE

Madeline Lombaerde
HP General Systems

A key element in isolating program problems (user or subsystem) is the proper interpretation of an abort message. This note deals with the principal methods for going from an abort message back to the line of source code where the abort occurred. The examples for FORTRAN and BASIC compiled programs use the methods that apply to SPL as well.

While it may seem as if the same methods can be applied directly to COBOL, there are certain conventions used by COBOL that are hidden from the user and significantly complicate the search for a program bug when the methods in this note are used. Therefore, the recommended procedure for debugging COBOL for now is to put in tracing DISPLAY statements, e.g., at the beginning of each paragraph and/or section. As for RPG, built-in debugging features should be used.

The following examples will show you two compiled programs and the steps required to identify the location of an error in the source code of each program when given an abort message.

NOTE

The figures in EXAMPLES 1 and 2 have been edited. Only data directly pertinent to the discussion of the examples is included.

Example 1 is a program (ABCINFO) compiled using FORTRAN A (figures 1A and 1B) and FORTRAN B (figures 2A and 2B).

Example 1: FORTRAN A and B

While running program ABCINFO, the following abort message occurs:

```
                    ABORT :$OLDPASS...%0,%13
                    PROGRAM ERROR #24: BOUNDS VIOLATION



    The program PMAP is:


PROGRAM FILE $NEWPASS.HLBOOK.MAL

ABC1                0
   NAME              STT   CODE ENTRY SEG
   ABCSETUP            1      0     0
   SEGMENT LENGTH             20
ABC0                1
   NAME              STT   CODE ENTRY SEG
   ABCINFO             1      0    11
   ABCSETUP            2                0
   TFORM'              3                ?
   FMTINIT'            4                ?
   TERMINATE'          5                ?
   IIO'                6                ?
   AIIO'               7                ?
   SEGMENT LENGTH            110

PRIMARY DB        0     INITIAL STACK     1440     CAPABILITY          600
SECONDARY DB      3     INITIAL DL           0     TOTAL CODE          130
TOTAL DB          3     MAXIMUM DATA     11610     TOTAL RECORDS         6
ELAPSED TIME   00:00:02.134                      PROCESSOR TIME   00:00.446
```

The source was compiled using Fortran A with the $CONTROL options of LABEL and MAP. Using Fortran B, $CONTROL LOCATION was used. See figures 1 and 2.

```
00001000  $CONTROL USLINIT,LABEL,MAP,SEGMENT=ABC0
00002000        PROGRAM ABCINFO
00003000        INTEGER ABC(100)
00004000      5 WRITE (6,600)
00005000    600 FORMAT(" NUM? ")
00006000        READ (5,*) NUM
00007000        IF (NUM.EQ.0) STOP
00008000        CALL ABCSETUP(ABC,NUM)
00009000     10 WRITE(6,601) ABC
00010000    601 FORMAT(10F6.2)
00011000        GO TO 5
00012000        END


  SYMBOL MAP

NAME              TYPE        STRUCTURE    ADDRESS

ABC               INTEGER     ARRAY        Q+  1,I
ABCSETUP                      SUBROUTINE
NUM               INTEGER     SIMPLE VAR   Q+  2


   LABEL MAP

  STATEMENT   CODE     STATEMENT   CODE     STATEMENT   CODE
    LABEL    OFFSET     LABEL    OFFSET     LABEL    OFFSET

      5         15        10        53      600 FMT      0
    601 FMT     5
```

Figure 1A.

```
00013000  $CONTROL SEGMENT=ABC1
00014000        SUBROUTINE ABCSETUP(I,N)
00015000        INTEGER I(N)
00016000        DO 1 J=1,N
00017000      1 I(J)=J
00018000      2 RETURN
00019000        END


  SYMBOL MAP

NAME              TYPE        STRUCTURE    ADDRESS

ABCSETUP                      SUBROUTINE
I                 INTEGER     ARRAY        Q-  5,I
J                 INTEGER     SIMPLE VAR   Q+  2
N                 INTEGER     SIMPLE VAR   Q-  4,I


   LABEL MAP

  STATEMENT   CODE     STATEMENT   CODE     STATEMENT   CODE
    LABEL    OFFSET     LABEL    OFFSET     LABEL    OFFSET

      1         11        2         15
```

Figure 1B.

```
PAGE 0001   HP32102B.00.00  FORTRAN/3000  (C) HEWLETT-PACKARD CO. 1976


00001000  $CONTROL USLINIT,LABEL,MAP,SEGMENT=ABC0,LOCATION
00015  00002000         PROGRAM ABCINFO
00015  00003000         INTEGER ABC(100)
00015  00004000       5 WRITE (6,600)
00032  00005000     600 FORMAT(" NUM? ")
00032  00006000         READ (5,*) NUM
00043  00007000         IF (NUM.EQ.0) STOP
00047  00008000         CALL ABCSETUP(ABC,NUM)
00053  00009000      10 WRITE(6,601) ABC
00074  00010000     601 FORMAT(10F6.2)
00074  00011000         GO TO 5
00075  00012000         END


  SYMBOL MAP

NAME              TYPE        STRUCTURE    ADDRESS

ABC               INTEGER     ARRAY        Q+  1,I
NUM               INTEGER     SIMPLE VAR   Q+  2
ABCSETUP                      SUBROUTINE

  LABEL MAP

 STATEMENT   CODE     STATEMENT   CODE     STATEMENT   CODE
   LABEL    OFFSET     LABEL    OFFSET     LABEL    OFFSET

     5         15        10         53      600 FMT      0

   601 FMT     5
```

Figure 2A.

```
PAGE 0002   HEWLETT-PACKARD 32102B.00.00   FORTRAN/3000


00004  00013000  $CONTROL SEGMENT=ABC1
00004  00014000        SUBROUTINE ABCSETUP(I,N)
00004  00015000        INTEGER I(N)
00004  00016000        DO 1 J=1,N
00011  00017000      1 I(J)=J
00015  00018000      2 RETURN
00016  00019000        END


  SYMBOL MAP

NAME              TYPE        STRUCTURE    ADDRESS

ABCSETUP                      SUBROUTINE
J                 INTEGER     SIMPLE VAR   Q+  2
I                 INTEGER     ARRAY        Q-  5,I
N                 INTEGER     SIMPLE VAR   Q-  4,I

  LABEL MAP

 STATEMENT   CODE     STATEMENT   CODE
   LABEL    OFFSET     LABEL    OFFSET

     1         11        2         15
```

Figure 2B.

The place to start is at the PMAP.

1. Find the name of the program unit in the segment %0 whose code begins before location %13 and whose last code location is greater than %13.

```
ABC1                          [0] ---> SEGMENT 0
   NAME                      STT    CODE  ENTRY  SEG
  [ABCSETUP]                   1       0      0
   SEGMENT LENGTH                      20                ) 13 IS BETWEEN 0 AND 20
```

Now we know the abort is in the subroutine ABCSETUP.
Next,

2. Calculate the relative program unit offset (RPUO) as follows:

   RPUO=Abort location – start of code

   From this formula, our RPUO for this example is

   %13 - %0 = %13

3. Now go to the compiler listing of ABCSETUP.

   a. Fortran A (Fig. 1b): Locate the statement label whose offset is less than %13 but such that the offset for the next label in the source code is greater than %13.

      The abort occurred between line 17 (stmt 1) and line 18 (stmt 2).

   b. Fortran B (Fig. 2b): Locate the statement whose relative code offset is less than %13 such that the offset for the very next statement is greater than %13.

      The abort occurred while trying to execute this statement.

      A bounds violation indicates that an attempt was made to address a location outside of the code or data statement. To check why the loop was going out of bounds, the user should check the data for NUM making sure that it is less than 100. If the data is OK, DEBUG can be used to check the value of J at the time of the abort (do a :SETDUMP and then check Q+2 in ABCSETUP).

Example 2 is a program (ABCBSC) compiled using BASIC.

Example 2: BASICOMP

While running program ABCBSC, the following abort message occurs:

```
                ABORT :$OLDPASS,,,%0,%367:SYSL,%154,%3741
                PROGRAM ERROR #24: BOUNDS VIOLATION

The program PMAP is

PROGRAM FILE $NEWPASS.HLBOOK.MAL

SEG'                0
   NAME              STT   CODE  ENTRY  SEG
   B'LLBL                     0      0
   B'ABORT


   TERMINATE'         12                  ?
   NOECHO              3    233    236
   ABCBSC              4    242    242
   B'INPUTNUM         13                  ?
   B'TERMIO           14                  ?
```

Example 2: BASICOMP (Continued)

```
   B'PRINTNUM          15                  ?
   B'STOP              16                  ?
   B'ABORTPLUSNUM      17                  ?
   SEGMENT LENGTH            424

PRIMARY DB        17     INITIAL STACK     1440     CAPABILITY          600
SECONDARY DB     424     INITIAL DL           0     TOTAL CODE          424
TOTAL DB         443     MAXIMUM DATA     11610     TOTAL RECORDS        11
ELAPSED TIME    00:00:03.708                        PROCESSOR TIME   00:00.621
```

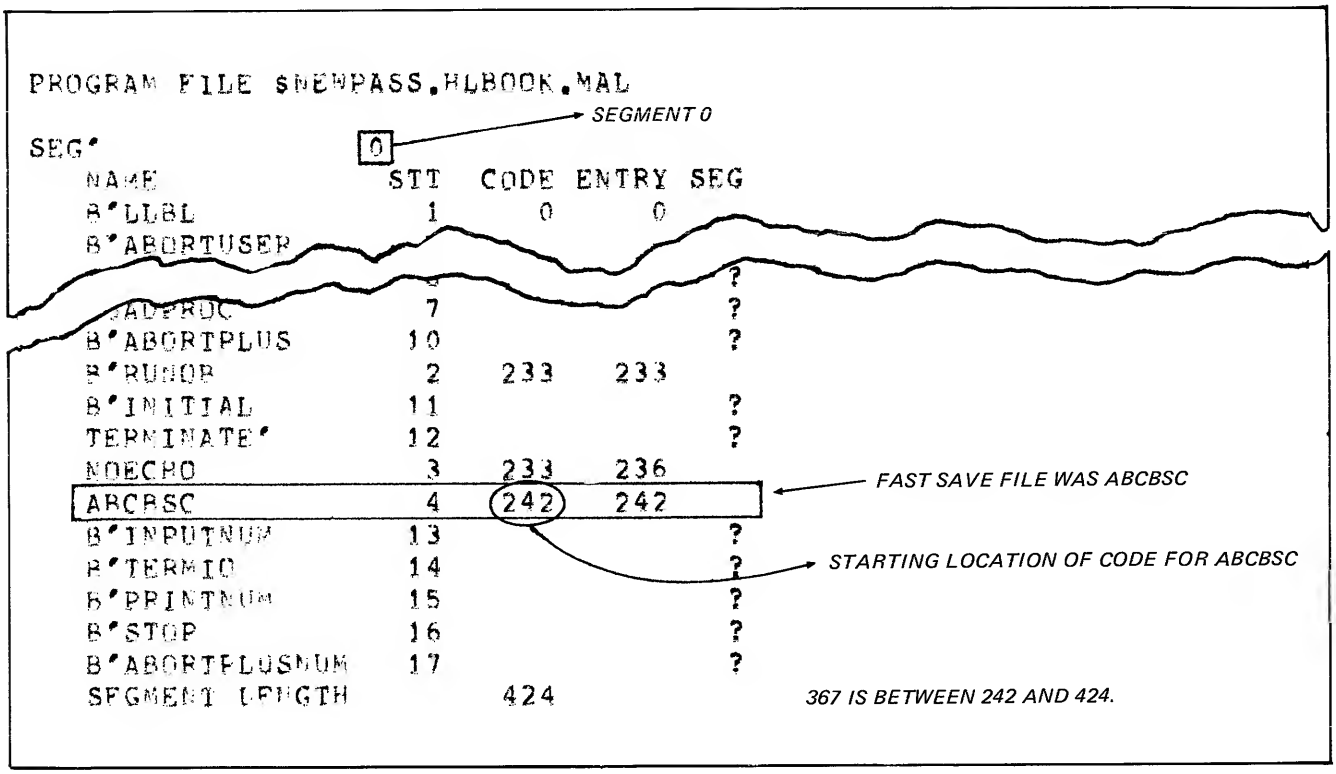
The Basic Fast Save file was compiled with $CONTROL SOURCE,LABEL,MAP

```
PAGE 0001    HEWLETT-PACKARD 32103B.P0.01(4WD)  BASICOMP


$CONTROL USLINIT,SOURCE,MAP,LABEL
$COMPILE ABCBSC
  10 DIM A[50]
  20 INPUT "I=",I
  30 IF I=0 THEN STOP
  40 FOR J=1 TO I
  50    A[J]=J
  60 NEXT J
  70 PRINT (FOR J=1 TO I,A[J])
  80 GOTO 20


    SYMBOL MAP

    ABCBSC

 NAME     TYPE            STRUCTURE         ADDRESS

  A       REAL            ARRAY             Q+  4,I
  I       REAL            SIMPLE VAR        Q+  5
  J       REAL            SIMPLE VAR        Q+  7


 LABEL MAP     ENTRY POINT      0    STACK    111
  LABEL LOCATION LABEL  LOCATION    LABEL  LOCATION    LABEL  LOCATION
    10= 20     41   20          41     30         54     40          61
    50         73   60         100     70        106     80=  20    136
```

Figure 3

Again, start with the PMAP:

1. Locate the program unit in Segment zero whose code begins before location %367 and whose last code location is greater than %367. This is in segment %0. The reason we go to %0.%367 instead of to location %154.%3741 is because this library (SL.PUB.SYS) routine is called at location %0.%367 and probably received bad data to begin with.



2. Now we know that the abort occurred in ABCBSC and we can calculate the RPUO as before:

   RPUO=Abort location – start of code

   RPUO=%367 – %242 = %125

3. Now go to the compiler listing (Fig. 3): look for the statement whose starting location is less than %125 but such that the next sequential statement has a beginning location greater than %125.

   The abort occurred during execution of statement 70. The data for I should be checked to assure that the value of I is less than or equal to 50.

bulletins

TECHNICAL CENTERS ESTABLISHED

Paul Myhre
HP General Systems

Effective November 1, 1976, the two GSD North American Customer Training Centers will be re-named Technical Centers. The Western Technical Center, (WTC) in Santa Clara, CA, and the Eastern Technical Center (ETC) in Rockville, MD, have been established to assure quality support services for their customers. The significance of the one-word change is that instead of being devoted solely to professional computer training at the Centers, their role will be expanded to include other support services, and customer training at the customer's site.

CUSTOMER TRAINING AT THE TECHNICAL CENTER

For most customers, this is the most economical method of training their staff. This is also the preferred and most efficient method for HP to train customers. Additional classes have been added to meet the increased demand for 3000 training and to bring the course availability down to 4 – 6 weeks.

CUSTOMER TRAINING AT THE CUSTOMER'S FACILITY

The WTC Staff will provide HP 3000 Comprehensive Introduction and System Management and Operation courses at the customer's site in Western North America. The ETC staff will provide the same courses for the Eastern North America customers.

The local Systems Engineers will continue to teach the one- and two-day courses on-site.

CX UPDATES

Mary Eicher
HP General Systems

If you have an HP 3000 CX System and are confused about updates, this issue of the **Communicator** is for you. We have separated CX and Series II information so that it is now easier for you to determine which manuals and which updates apply to your system. If you have an HP 3000 CX System read down the *left* side of the documentation lists; if you have a Series II System read down the *right* side of the list.

In the future we shall convert more of the existing manuals for use with either a CX or Series II System. Updates to these manuals will be listed in the documentation section of the **Communicator** under the system to which they apply.

software updates

Each issue of the **Communicator** provides you with information pertinent to the status of 3000 software products including the latest software changes and enhancements.

The 3000 software updates described in this issue relate to the following products:

PRODUCT	3000 SERIES I	3000 SERIES II	NUMBER	UPDATE AND FIX LEVEL	MIT TAPE DATE CODE
MPE I	X		32000C	00.13	1636
MPE II		X	32002A	00.04	1640
BASIC/3000	X	X	32101B	00.04	1640
COBOL/3000	X		32213B	02.02	1640
COBOL/3000	X	X	32213C	01.00	1634
Compiler Library/3000	X		32211C	04.04	1640
Compiler Library/3000		X	32211D	00.04	1640
DEL/3000		X	32206A	01.01	1640
EDITOR	X	X	32201A	05.01	1640
Fortran/3000	X	X	32102B	00.04	1640
RPG/3000	X	X	32104A	02.08	1640

Where changes in documentation are indicated, updates to the appropriate manuals will be printed. This information is provided simply as a temporary measure.

MPE 32000C.00.13 AND MPE 32002A.00.04

This article describes MPE 32000C.00.13 and MPE 32002A.00.04 as incorporated into the 3000 Series I and 3000 Series II MIT tapes date coded 1636 and 1640. The information in the article is organized as follows.

1. Modules modified for MPEI C.00.13 and MPEII A.00.04
2. Supported Utility Module changes for MPEII A.00.04
3. List of problems solved in MPEI C.00.13 and MPEII A.00.04
4. Enhancements to MPEII A.00.04
5. Known problems in MPEII A.00.04

1A MODULE CHANGES		C.00.XX		MPE C.00.13									
MODULE		1	2	3	4	5	6	7	8	9	10	11	12	13
INITIAL	0	X		X	X			X	X	X	X	X	X	X
SYSDUMP	1	X	X	X			X		X	X			X	
SEGPROC	2	X	X				X			X	X			
SEGDVR	3												X	
DISPATCH	4			X			X				X	X	X	
LOAD	5		X										X	
MAPP	6					X								
UCOP	7	X												
DEVREC	8													
PROGEN	9	X							X	X	X		X	
IMIN	10					X		X			X	X		X
EXIN	11	X	X	X		X	X	X		X		X		
LOG	12	X									X			
IOPTRD0	13													
IOPTPH0	14						X		X					
IOPLOT0	15													
IOMDISK0	16			X				X	X	X				
IOFDISK0	17			X				X	X					
IOTAPE0	18				X				X					
IOLPRT0	19												X	X
IOCDRD0	20		X				X							
IOCLTTY0	21													
IOTERM0	22											X		
IOCDPN0	23													
IOPRPN0	24					X	X					X	X	
IOREM0	25													
IOBSC0	26													
IOMDISK1	27	X		X				X	X	X	X			
PFAIL	30			X	X	X								
FILESYS	50	X	X	X	X	X	X	X	X	X	X	X	X	X
COMM' INT	51	X		X			X					X	X	
STORE/RESTORE	52			X	X			X	X	X			X	
DIRC	53									X				X
ALLOCATE	54		X		X				X			X	X	
DISKSPC	55	X												
MMCORER	56						X		X			X		
MMDISKR	57											X		
ABORTRAP	58						X	X		X		X		X
MESSAGE	59							X		X	X	X		
CROUTINE	60			X	X						X	X		
IOUTILITY	61	X		X	X			X	X	X				
TTYINT	62		X	X	X			X		X				
PCREATE	63	X									X			
MORGUE	64			X						X	X			
PROCMAIL	65													
PINT	66						X		X	X		X		
DATASEG	67	X								X				
IOPM	68		X			X						X	X	
CHECKER	69													
UTILITY	70	X	X	X		X					X			X
SEGUTIL	71	X		X				X		X			X	
LOADER1	72		X	X					X		X			

1A MODULE CHANGES C.00.XX MPE C.00.13 (continued)													
MODULE		1	2	3	4	5	6	7	8	9	10	11	12	13
RINS	73						X						X	
JOBTABLE	74	X												
DEBUG	75	X												
NURSERY	76			X										
SYSDPLY	77						X							X
FIRMWARESIM	78	X								X	X	X		
SPOOLING	79			X	X			X	X	X	X	X		
SPOOLCOMS	80	X					X		X		X		X	
MESSAGE CAT	–			X			X	X	X	X	X		X	X

1B. MODULE CHANGES A.00.XX					
MODULE		1	2	3	4
INITIAL	0		X	X	X
SYSDUMP	1		X	X	X
SEGPROC	2		X		
SEGDVR	3		X		
DISPATCH	4		X		
LOAD	5		X		
UCOP	7		X		X
DEVREC	8				X
PROGEN	9		X	X	X
ININ	10				X
MEMLOGP	11		X		X
LOG	12				X
IOPTRD0	13		X	X	
IOPTNP0	14		X	X	X
IOPLOT0	15				
IOMDISC0	16				
IOFDISC0	17				
IOTAPE0	18				
IOLPRT0	19		X		X
IOCDRD0	20				
IOTERM0	22		X	X	X
IOPRPN0	24				
IOREM0	25				
IOMDISC1	27				
PFAIL	30				X
FILESYS	50		X	X	X
COMM'INT	51			X	X
STORT/RESTORE	52				
DIRC	53				
ALLOCATE	54				
DISCSPC	55				
MMCORE	56				X
MMDISKR	57		X	X	X
ABORTTRAP	58		X		
MESSAGE	59				X
CROUTINE	60				
NRIO	62		X	X	

1.B MODULE CHANGES (Continued)					
MODULE		1	2	3	4
PCREATE	63		X		
MORGUE	64				
PROCMAIL	65				
PINT	66		X		
DATASEG	67		X		X
CRIO	68		X	X	X
CHECKER	69				
UTILITY	70				X
SEGUTIL	71		X		
LOADER1	72		X		
RINS	73	X		X	
JOBTABLE	74				
DEBUG	75				X
NURSERY	76				X
STKDUMP	77				
FIRMWARESIM	78				
SPOOLING	79				
SPOOLCOMS	80				
MESSAGE CAT	–				

2. SUPPORTED UTILITY MODULE CHANGES FOR MPEII A.00.04					
MODULE		1	2	3	4
DISKEDT2					
DPAN2			X	X	X
FREE2					
LISRDIR2					
LISTEQ2					
LISTLOG2			X	X	
PATCH2					
MEMLOGAN					
MEMTIME					
SADUTIL					
SLPATCH					

3A. PROBLEMS SOLVED IN MPE C.00.13

a. In some cases, disc space for the system disc was not being returned to the system during system initialization. This could result in the "OUT-OF-SYSTEM-DISC" situation. This has been corrected.

b. The parameter at Q+1 of the internal interrupt procedure parity error was being destroyed while scanning memory for the memory location in question. Thus losing the exact cause (data, memory, or system) of the interrupt.

c. The line printer driver IOLPRT0 has been modified to enable some previously disabled line printer dependent interrupts.

d. A check, during file opening, for valid serial output device types has been corrected to include the card reader punch, type 37.

e. The condition code from the procedure DIRECSETFLAG was not being returned properly in one obscure case. This has been corrected.

f. A stack adjustment for traps resulting from regular decimal FIRMWARESIM was not being properly performed. This has been corrected.

g. Form feeds to some CRTS will erase the screen. To preserve pertinent displayed information during some aborts, page ejects for stack dump headings will be suppressed to terminals when running from a session.

h. Input arguments to the intrinsic binary of 65536, 65537, 65538, and 65539 will now return overflow.

3B. PROBLEMS SOLVED IN MPEII A.00.04

a. The 2892 card reader can now be spooled in.

b. The GETPRIORITY intrinsic can now be called with the last PIN in the PCB.

c. You can no longer free a RIN that does not belong to your process.

d. Under certain conditions, Control Y was causing a stack overflow problem that has now been resolved.

e. The CTRANSLATE intrinsic now returns condition codes CCE/CCG.

f. The BINARY intrinsic now returns condition code CCG for 65536 through 65539.

g. :EOD now terminates :DATA stream when the terminal is opened for I/O access.

h. Column binary read is now allowed with the 2892 card reader.

i. The paper tape punch will no longer truncate the odd byte in binary mode while using system buffers.

j. Opening more than 4 extra data segments per process will work now.

k. SYSDUMP was modified to handle 0 to 63 code segments per process.

l. A deadlock was resolved where a SON process was trying to access a resource that the FATHER process had.

m. MEMLOGP and LOG can not be run without SM and PM capabilities. It should not be run even if you have this capability unless you want to lock out your terminal.

n. DISABLE/ENABLE has been added to SYSIOPROC. This fixes a problem of SYSIOPROC occasionally not servicing a request.

o. A SF124 will now occur if a call is made to RELDATASEG with a DST# <= %63.

p. When getting space in virtual memory, INITIAL now cuts back the stack properly avoiding a potential stack overflow.

q. Fixed problem which caused certain errors to display the wrong error message.

r. Debug has been fixed such that "E@" now closes list file and "B@" now displays correct LCST#.

s. Open MEMLOG has been modified to zero both MEMLOG file records when creating new MEMLOG file.

t. The line printer driver has been modified to correct a situation where a "SPOOLEE I/O ERROR" could be caused by toggling the ON LINE/OFF LINE switch.

u. Two no-memory related problems were resolved. Unblocked I/O requests now retry, and ZSIZE/stack overflow situations are handled correctly now.

v. LOAD has been modified such that system modules prepared in a normal mode cannot be accessed by any user.

For example: users cannot :RUN LOAD.PUB. SYS or other system files.

4. ENHANCEMENTS TO MPEII A.00.04

a. The :APL command has been added. This command causes the APL subsystem to be invoked, if it is present in PUB.SYS. It has no parameters.

b. PFAIL now calls initialization PLABEL in DLT on auto restart.

c. **COMMAND LOGON FACILITY**

The special requirements of APL terminals have resulted in a minor, but useful, enhancement to MPE II. The primary problem was that APL uses a couple of special encodings to support its unique character sets. When a user on one of the terminals logged on and off, garbage messages would be produced by MPE. The solution chosen was to have the Command Inter-

preter detect the character sets and translate the garbaged messages into the appropriate APL encoding. At the request of the APL project, the scheme was further enhanced such that the APL subsystem was entered immediately upon logon, without user intervention, and the user was logged off immediately upon exiting the subsystem. It was seen that this feature could be of value to others besides APL users and so was generalized into the following scheme.

Two types of logon are available to interactive users:

Normal: HELLO *user.account* — etc. —

New: (*cicommand*) *user.account* — etc. —

Where *cicommand* is any legal CI command string. DEVREC validates the user, passes the character string contained within the parentheses to the CI which then executes the command. The CI, immediately upon resuming control, performs a logoff. All outputs from the CI are listed to the terminal as usual. If one of the special APL character sets is detected, that subsystem is notified through an appropriate parm value.

Special Rules and Restrictions:

1. Break is enabled.
2. Users who log on using one of the special APL character sets will only be allowed to invoke the APL subsystem.
3. Commands which would normally cause an end of file, such as :HELLO, :BYE, :JOB, :DATA, :EOF, :EOJ, will not be recognized and the session will be immediately logged off.
4. Leading and trailing blanks will be suppressed.
5. This is only allowed from sessions, not jobs.
6. The special logon sequence is not recognized within sessions or jobs, as, for example, :HELLO is.

EXAMPLES OF USE

```
:(SHOWJOB) MANAGER.SYS
:(LISTF.2) BOB.TP
:(APL) JEUNG.LANG,QUAD
```

Potentially very useful is the following

```
:(RUN FCOPY.PUB.SYS)
OLSON.FIELD.SUPPORT,HP32102
  <<logon message followed by FCOPY header
     followed by
>    <<FCOPY prompt. Now hit break>>
:FILE LP;DEV=FASTLP
:RESUME
READ PENDING
FROM=XYZ;TO=*LP
EXIT
<<logoff message>>
:(STREAM) BOB.XYZ
>!JOB XXX,BOB.TP
>!SPL A,B,C,D,E
>!EOJ
#J1234
>:
<<logoff message>>
```

5A. KNOWN PROBLEMS IN MPE C.00.13

a. Closing a tape file with no rewind is not implemented.

b. FSPACE spaces tape files by blocks rather than by records.

c. Chained SIOS on magnetic tape do not perform correctly, causing transfer of blocks larger than 4096 words to fail if the record format is variable or undefined.

d. The character ":" is treated as an EOF on $STDINX.

e. The commands: LISTACCT, LISTGROUP, and LISTUSER can lock the directory indefinitely if the output is written to magnetic tape and the tape is not ready.

f. If the FORMSG parameter of FOPEN begins on an odd byte boundary, the preceding byte is also printed.

g. Lower case :EOD is not recognized as an end-of-file on data accepting devices.

h. Issuing a :DEALLOCATE command for a non-existent program file returns an error 217. The error should be error 217,52. The 52 is the file system error number returned from FCHECK.

i. Debug break points cannot be set in dynamically loaded procedures except by specifying the physical CST numbers.

j. When DPAN finds that the PCB table has been filled, it prints the erroneous messages "INVALID FIRST UNASSIGNED ENTRY" and "INVALID BACKWARD SUBQUEUE POINTER" even though there is no error in the PCB.

k. When the maximum number of open SPOOFLES is not sufficient to handle all spooling requirements, spooled jobs may cause endless numbers of null list files to be generated. This bug manifests itself as multiple $STDLIST files for a single job, each producing only a header and trailer. If the line printer is spooled, this results in many null SPOOFLES, each using four sectors of disc space. If the line printer is not spooled, these null SPOOFLES will begin printing immediately unless the printer is not ready. In this case, the system will crash due to an IOQ overflow. If an open SPOOFLE is closed during this resource allocation loop, the job may be launched normally.

In this case, the last SPOOFLE for $STDLIST will be the true job listing.

This bug can be overcome by increasing the maximum number of open SPOOFLES. The recommended value is 20, but a more exact figure can be found by examining the usage of your system. Each initial allocation (FOPEN) of a spooled device uses one open spoofle. When the file is closed (FCLOSE), the spoofle becomes unopened.

For example:

> A session's single access to a spooled line printer requires one opened SPOOFLE; a spooled job requires at least two, one for $STDIN and one for $TDLIST. Each additional access to a file of device class LP requires an additional open SPOOFLE.

One indication that the limit is being reached is allocation failures for spooled devices.

5B. KNOWN PROBLEMS IN MPEII A.00.04

SYSTEM FAILURE TYPE:

a. SF130 – Memory manager loses a MTAB entry. If enough are lost a SF130 will occur.

b. SF311 – "Abort while critical". Occurs when a code segment or memory links are getting destroyed.

c. SF30 – Under a very heavy load, one can get an SF30 while switching Log files.

d. SF206 – "Invalid LDEV for ATTACHIO". This is an intermittent problem with very little data.

FUNCTIONAL FAILURE TYPE:

1. A session with an outstanding READ cannot be aborted until reception of a carriage return.
2. A call to XARITRAP after a call with an illegal PLABEL returns garbage to OLDLABEL.
3. Enabling of logging record 8 is supposed to lower default OUTPRI to 8.

BASIC/3000, HP 32101B.00.04

This article describes BASIC/3000, HP 32101B.00.04 as incorporated in the 3000 Series I and Series II MIT tape date coded 1640.

PROBLEMS CORRECTED IN BASIC/3000:

1. Specifying "END" on a file PRINT statement to an ASCII device which cannot write an end-of-file (e.g., $STDLIST) resulted in a superfluous error message.
2. If an I/O error occurred on a binary file when executing the "END" function on a file PRINT statement, an incorrect error message was given.
3. After reporting a file I/O error (e.g., end-of-file), the file was sometimes not closed.
4. I/O errors (e.g., end-of-file) occurring when writing to binary files were not reported in a timely manner because detection of these errors was deferred until some I/O to a different record in the same file was detected. This has been fixed for files with fixed- and undefined-length records; however, no fix is possible for files with variable-length records. Note that this problem does not affect reading from a binary file or any I/O to ASCII files (regardless of the record type) since the physical I/O in these cases coincides with the logical I/O.

Note that the user can utilize the LIN function to control when the physical write occurs to binary files with variable-length records. Thus, the programmer can force the physical write to coincide with the last logical write to a record and thereby force detection of file errors in a timely manner.

The problem cannot be fixed for writes to binary files with variable-length records without altering the results of I/O to binary files or without reinstating a restriction imposed by BASIC prior to version B.0.3, to wit: the BASIC program could not access records beyond the physical record limit. Since MPE permits the number of logical records to exceed the physical record limit, BASIC cannot anticipate the end-of-file when performing some logical I/O to the file. Consequently, detection of the end-of-file is deferred until the physical write to the file, caused by accessing another record in the file or by closing the file.

KNOWN PROBLEMS IN BASIC/3000:

1. The interpreter aborts with a stack underflow when control-Y is typed in certain circumstances. This occurs most often when printing the FREQ table. The problem may also arise in some cases when INVOKing or using the ABORT, CALLS or FILES commands in BREAK-mode. [BP #1396]

   Work-around (for BREAK-mode commands): Type control-Y and set a breakpoint at the next statement to be executed. Then enter the GO or RESUME command. When you break at the next statement, it will then be safe to use any BREAK-mode commands.

COBOL/3000, HP 32213B.02.02

This article describes COBOL/3000, HP 32213B.02.02 as incorporated in the 3000 Series I MIT tape date coded 1640.

PROBLEMS CORRECTED IN BASIC/3000:

1. A GO TO statement to the first paragraph in a subprogram did not work properly.
2. A RENAMES clause was erroneously being detected as an error if there were any OCCURS clauses in the record. Now only the following OCCURS clause conditions are errors in RENAMES clauses:
   a. The item being renamed is subordinate to an OCCURS clause.
   b. The item being renamed contains an OCCURS clause.
   c. The item being renamed has a subordinate item containing an OCCURS clause.
3. An illegal RENAMES clause sometimes caused the compiler to loop or abort.
4. Compiling into a non-empty USL file which was built with more than 2040 records sometimes caused the USL file to be initialized even though $CONTROL USLINIT was not specified.
5. Compiling a subprogram containing secondary entry points with parameters into an non-empty USL file did not inactivate previous entry points with the same name.
6. Compiling programs into a non-empty USL file sometimes caused the compiler to loop. This problem usually occurred only when secondary entry points were present in the program being compiled.
7. The construct IF A=B AND (C+D>0) caused an extra word to be deleted from the stack at run time.
8. Producing a symbol table map ($CONTROL MAP) for a Data Division with several hundred items with the same level number subordinate to the same item caused a stack overflow.
9. Illegal duplicate paragraph names caused the compiler to loop.
10. Using a COMP-3 item as a subscript did not work properly.
11. Moving numeric data into an alphanumeric field (PIC X) in a table did not work properly.
12. The COBTEMP file filled up during compilation, thereby aborting the compiler.

CHANGES MADE IN THE COMPILER LIBRARY FOR COBOL, HP 32211C.04.03:

The following problem has been fixed in the COBOL Library version 4.03:

A SEARCH ALL statement could occasionally find a match just after the end of the table being searched.

The following enhancement has been added to the COBOL Library in version 4.03:

Files assigned to terminals can now be opened for I-O providing a :FILE command specifying ACC=INOUT has been executed. This capability does not apply to $STDIN or $STDLIST.

COBOL/3000, HP 32213C.01.00

This article describes COBOL/3000, HP 32213C.01.00 as incorporated in the 3000 Series I and Series II MIT tape date coded 1634.

This is the initial release of the COBOLC compiler. The compiler is basically the same as the COBOLB compiler except that it can compile larger programs. The Symbol Table will be extended to disc as soon as it gets too large for available memory.

The generated code is the same as for the COBOLB compiler.

The COBOLC.01.00 compiler is at the same level as the COBOLB.02.01 compiler except for the following additional corrections:

1. Multiply defined symbols are now correctly diagnosed. They could result in looping of the COBOLB compiler.
2. MOVE of numeric items to alphanumeric table elements are now correctly executed.

The following internal files are used in the compiler:

1. COBTEMP.
   File existed in COBOLB compiler. It passes intermediate code to the code generating phase.
   Record size=128 words, file size/no.of extents=default.
2. COBSTAB.
   Contains symbols-part of Symbol Table. It exists with COBDTAB, only if the Symbol Table is extended to disc. Record size=256 words, file size=512, no. of extents=16. In general the file con-

tains 240 records. Any additional records are added one at a time when needed.

3. COBDTAB.
Contains data-part of Symbol Table. It exists with COBSTAB, only if the Symbol Table is extended to disc. Record size=1024 words, file size=128, number of extents=16. In general the file contains less than 32 records. Any additional records are added one at a time when needed.

KNOWN ERRORS IN COBOL/3000:

1. GO TO or PERFORM first paragraph in a subprogram without sections may cause Bounds Violation at object time.
2. Certain invalid RENAMES may cause Bounds Violation at compile time.

COMPILER LIBRARY/3000, HP 32211C.04.04

This article describes Compiler Library/3000, HP 32211C.04.04 as incorporated in the 3000 Series I MIT tape date coded 1640. (This version of the Compiler Library only supports three word long floating-point operations.)

PROBLEM CORRECTED IN THE COMPILER LIBRARY/3000:

A bug was introduced in version C.00.02 such that when a write was made of a length greater than the logical record length of the output file, the FORTRAN formatter went into an infinite loop. This problem in version C.04.03 was fixed in this version:

COMPILER LIBRARY/3000, HP 32211D.00.04

This article describes Compiler Library/3000, HP 32211D.00.04 as incorporated in the 3000 Series II MIT tape date coded 1640. (This version of the Compiler Library only supports four word long floating-point operations.)

PROBLEM CORRECTED IN COMPILER LIBRARY/3000:

A bug was introduced in version D.00.02 such that when a record was written which was longer than the logical record length of the output file, the FORTRAN Formatter went into an infinite loop. This has been corrected.

DEL/3000, HP 32206A.01.01

This article describes DEL/3000, HP 32206A.01.01, as incorporated in the 3000 Series II MIT tape date coded 1640.

PROBLEMS CORRECTED IN DEL/3000:

1. A bug has been removed from the procedure READTERM, it no longer destroys the byte beyond the end of the input buffer.
2. The procedure GETFORM will now return an error code of -4 if the users buffer length is less than 64. Prior to this change GETFORM did not return the form definition if the buffer length was less than 64, but it did not notify the program of this condition.
3. FORMAINT's form modification procedure has been changed: a form being modified will no longer be lost when the form modification is interrupted.
4. The procedure NEXTEDIT will now return all edits specified for the last field of a form. Prior to this fix it would only return 1 edit description for the last field of a form.

EDITOR/3000, HP 32201A.05.01

This article describes EDITOR/3000, HP 32201A.05.01, as incorporated in the 3000 Series I and Series II MIT tape date coded 1640.

PROBLEMS CORRECTED IN EDITOR/3000:

1. The token FIRST used as a column position indicator, e.g., (FIRST), will indicate the first non-blank character of an incumbent record.
2. The token LAST used as a column position indicator, e.g., (LAST), will indicate the last non-blank character of an incumbent record, rather than the first evaluation of a column position within a range.
3. These commands are now indicated as invalid:
NQ, NOTQ, OQ, ORQ, PQ, PROCEDUREQ, SQ, SETQ, TQ, TEXTQ, UQ, USEQ, VQ, VERIFYQ, WQ, WHILEQ, XQ, XPLAINQ.
These commands had never been valid, in that the Q suffix had no meaning associated with it.
4. Variable length record files may be kept, texted, and joined. Note that SET variable must be used. Note also that the JOIN command will elide line number characters unless the UNN option is used.

5. The LEFT, RIGHT, and LENGTH values are now retained in files created by the KQ and KEEPQ command, as well as in K files.
6. Error message 29, INVALID LINE NUMBER, now will display the record number associated with the error properly.
7. CHANGE command oldstrings with trailing blanks now will match with lines containing the oldstring at the end of a line.
8. CHANGE command inadvertent line deletion has been corrected. The objective is to delete those changed lines which become empty.
9. CHANGE command line deletion no longer deletes a portion of the line following.
10. The FIND command upper bound, whether in its implied form or in its explicitly stated form in a range, is now computed properly.
11. JOIN computation of available space has been corrected.
12. Keeping a file across accounts to an existing file will now result in an error. The KEEP file will not be purged. The command will not be executed. Note that in general when read access is granted it is generally possible otherwise to execute a programmatic purge of a file across accounts.
13. KEEPQ or KQ to a file will now issue a warning message upon successful completion that the work file is undefined. The message will be output only in long mode, and not when short has been set.
14. KEEP file security testing has been refined such that only errors 92 and 93 are regarded as violations.
15. LIST command pointer setting at the conclusion of a range is now at the SET RIGHT value rather than at the physical end of the record.
16. MODIFY ALL may now be in either lower or upper case.
17. MODIFY command range list element parsing has been corrected.
18. PROCEDURE names may be in lower case as well as upper case.
19. PROCEDURE command range list element parsing has been corrected.
20. PROCEDURE command SL file identification is now emitted as an 'S', A 'P' OR A 'G' after the procedure name when error 35 is issued.
21. Q command without a delimited string will act as a line space.
22. SET SIZE is now defined as the number of lines of text needed in the work file instead of an absolute number of records.
23. SET SHORT will now inhibit echoing commands in jobs. Heretofore command echoing was inhibited by the SET SHORT only in WHILE command command blocks.
24. Text of a file created by a KEEPQ or KQ command or a K file will issue a warning message that the text file is now the work file. This warning may be disabled by a SET SHORT.
25. VERIFY TIMES is now parsed without error.
26. XPLAIN JOIN and XPLAIN J will now produce identical results.

OUTSTANDING PROBLEMS IN EDITOR/3000:

1. Use of the CONTROL Y during a modify is ambiguous: it may be interpreted to restart the process using the original line value; also, it may be interpreted to exit from the command.
2. Re-setting the SIZE value may inhibit texting a file in its entirety.
3. KEEP across accounts to a non-existent file will result in a file system security error.
4. Some SET parameters are at default values when WORK files (K FILES) are texted. This is also true of files created using the KEEPQ or KQ command.

FORTRAN/3000, HP 32102B.00.04

This article describes FORTRAN/3000, HP 32102B.00.04, as incorporated in the 3000 Series I and Series II MIT tape date coded 1640.

PROBLEMS CORRECTED IN FORTRAN/3000:

1. Logical expressions which contained a type conversion of a logical expression from logical to integer through use of the intrinsic INT would have an erroneous BRE emitted.

   For example, the statements

   ```
   LOGICAL STATUS
   IF (INT(STATUS).NE.0) STOP 1
   ```

   would result in bad code.
2. The IO list in a DISPLAY statement could not start with a real, double precision or double integer constant. (BR807)
3. Fortran would not accept the initialization of both a real and a double integer constant in the same data statement.

4. String compares where the strings were of different lengths and were compared using the .EQ. operator always compared as equal, completely contrary to the definition in the manual.
5. The use of a parameter in a procedure call of the form /-nnn/, where nnn is a number between 1 and 255 caused the system intrinsic processor to be invoked accidentally. The problem always manifested itself as a bounds violation in the compiler near %0.%6072.
6. In certain circumstances, calling a system intrinsic through the SYSTEM INTRINSIC statement would result in the SEGMENTER detecting an error in the way the compiler handled the option variable mask of option variable system intrinsics.

RPG/3000, HP 32104A.02.08

This article describes RPG/3000, HP 32104A.02.08, as incorporated in the 3000 Series I and Series II MIT tape date coded 1640.

PROBLEMS CORRECTED IN RPG/3000:

Fixes for 02.08 (C) indicates a change to the compiler and (L) indicates a change to the library.

1. (C) An index used before it was defined resulted in incorrect code being generated.
2. (L) TESTN sometimes aborted with a bounds violation.
3. (C) A tag name that was defined in an RPG program was listed as an unresolved external at program load time.
4. (L) Multiple USWITCH records were not accepted.
5. (C) (L) Fetch overflow sometimes resulted in duplicate printing of overflow lines.
6. (L) RSAM update files did not properly do an update immediately following an ADD. The record just added was not updated.
7. (L) User switches did not work properly with IMAGE files.
8. (C) File translation tables supplied by the user gave invalid error messages and did not work properly.

ENCHANCEMENTS TO RPG/3000:

1. (C) When compiling into an old USL file, outer blocks and entry points are now ceased before the new ones are put in. Therefore, USLINIT is no longer needed all the time on multiple compiles.
2. (C) A seven digit packed input field that is edited for output with a Y edit code will print as though it were 6 digits.

documentation

The following tables list currently available customer manuals for HP 3000 products. This list supersedes the list in the last issue of the **Communicator**.

The most recent changes to the tables are indicated for easy reference. Prices are subject to change without notice.

Copies of manuals and updates can be obtained from your local Sales and Service office. The address and telephone number of the office nearest to you are listed in the back of all customer manuals.

Update packages are free of charge. If you require an update package complete the Update Order Form in the back of the **Communicator** and mail the form to:

Software/Publications Distribution
5303 Stevens Creek Blvd.
Santa Clara, Ca. 95050

Customers in the U.S. may also order directly by mail. Simply list the name and part number of the manual(s) you need on the Corporate Parts Center form supplied at the back of the **Communicator**.

A few words about documentation terms:

New	A new manual referes only to the first printing of a manual. When first printed, a manual is assigned a part number.
Revised	A revised manual is a printing of an existing manual which incorporates new and/or changed information in its contents. For example, a manual is revised when an update package is incorporated into the manual: the manual gets a new print date and the update package disappears. Note that a revision to a manual effectively obsoletes the previous version of the manual.
Update	An update package is a supplement to an existing manual which contains new and/or changed information. Updates are issued when information must get to customers, yet it is inappropriate to issue a revised manual. An update has no part number, it is automatically included when you order the manual with which it is associated.

MPE/3000 MANUALS

SERIES I (CX)					SERIES II	
PUBLICATION DATE	UPDATE	MANUAL TITLE	PART NUMBER	PRICE	PUBLICATION DATE	UPDATE
		Console Operator's Guide	30000-90013	$ 7.00	6/76	
1/75	10/75	Console Operator's Guide, 32000C MPE/3000	32000-90004	7.00		
		Error Messages and Recovery Manual	30000-90015	17.00	6/76	
		General Information Manual	30000-90008	6.50	10/76*R	
11/73		General Information Manual, Multiprogramming Executive	03000-90096	4.00		
		HP 3000 CX to HP 3000 Series II Program Conversion Guide	30000-90046	4.00	6/76	
		Index to MPE Reference Documents	30000-90045	5.50	6/76	
1/75	6/76	MPE/3000 Reference Manual, 32000C	32000-90002	19.50		
		MPE Commands Reference Manual	30000-90009	12.50	6/76	
		MPE Intrinsics Reference Manual	30000-90010	15.00	6/76	
		MPE Segmenter Reference Manual	30000-90011	4.00	6/76	
		MPE Debug/Stack Dump Reference Manual	30000-90012	7.50	9/76	9/76†
		MPE System Utilities Reference Manual	30000-90044	5.00	6/76	
10/75		MPE/3000 Operating System, System Utilities Manual	32000-90008	3.00		
		Software Pocket Guide	30000-90049	3.50	6/76	
7/75		Software Pocket Guide, HP 3000	03000-90126	3.50		
		System Manager/System Supervisor Manual	30000-90014	10.00	6/76	9/76
10/75		System Manager/System Supervisor Manual, 32000C MPE/3000	32000-90006	14.00		
6/75		Using the HP 3000: A Guide for the Terminal User	03000-90121	7.50	6/75	

*R = Revised Manual

LANGUAGE MANUALS

SERIES I (CX)					SERIES II	
PUBLICATION DATE	UPDATE	MANUAL TITLE	PART NUMBER	PRICE	PUBLICATION DATE	UPDATE
		BASIC Interpreter Reference Manual	30000-90026	$11.50	6/76	
7/75		BASIC/3000 Interpreter Reference Manual	03000-90008	10.00		
9/74		BASIC/3000 Interpreter Pocket Guide	03000-90050	2.50		
11/74		BASIC/3000 Compiler Reference Manual	32103-90001	3.50	11/74	6/76
11/72		BASIC for Beginners	03000-90025	5.50	11/72	
7/75		COBOL/3000 Reference Manual	32213-90001	12.50	7/75	6/76
5/76		Cross Assembler for 2100 Computers Reference and Application Manual	03000-90047	12.00	5/76	
		FORTRAN Reference Manual	30000-90040	9.50	6/76	
3/76		FORTRAN/3000 Reference Manual	32102-90001	13.50		
2/75	8/76	RPG/3000 Compiler Reference and Application Manual	32104-90001	16.50	2/75	8/76
4/75		RPG Listing Analyzer	32104-90003	0.50	4/75	
11/73		SPL/3000 Reference Manual	03000-90002	7.50		
11/73		SPL/3000 Textbook	03000-90003	13.00		
		System Programming Language Reference Manual	30000-90024	15.00	9/76	9/76†
		System Programming Language Textbook	30000-90025	11.00	6/76	9/76

†*There are instances where a new edition of the manual and an update are both shown with the same date. The new edition incorporates all changes made in the update. If you already have the manual, order the update only; otherwise order the new edition.*

ADDITIONAL MANUALS

SERIES I (CX)		MANUAL TITLE	PART NUMBER	PRICE	SERIES II	
PUBLICATION DATE	UPDATE				PUBLICATION DATE	UPDATE
		2780/3780 Emulator Reference Manual	30000-90047	$ 8.50	6/76	
12/74	2/76	2780/3780 Emulator Subsystem Reference and Application Manual	30103-90001	10.00		
		Compiler Library Reference Manual	30000-90028	12.00	6/76	
2/76		Compiler Library Reference Manual, HP 3000	03000-90009	16.50		
		Data Entry Library Manual	30000-90050	6.50	6/76	
8/75		EDIT/3000 Reference Manual	03000-90012	7.50	8/75	6/76
6/76		FCOPY Reference Manual	03000-90064	6.00	6/76	
6/76		HP 2894A Card Reader Punch Operating and Programming Manual	30119-90009	7.00	6/76	
10/74		HP 3000 Cross Loader for HP 2100 Computer	03000-90107	11.00	10/74	6/76
2/75	5/75	IBM 1130/1800 to HP 3000 FORTRAN Conversion Guide	36995-90013	6.00		
12/75		IBM System/3 to HP 3000 Conversion Guide	32104-90004	7.50		
		IMAGE Data Base Management System Reference Manual	30000-90041	4.50	6/76	
3/76		IMAGE/3000 Reference Manual	32215-90001	7.00		
		Instruction Decoding Pocket Guide	30000-90057	1.00	6/76	
		Line Printer Operating and Programming Reference Manual	30209-90008	6.00	6/76	
		Machine Instruction Set Reference Manual	30000-90022	7.00	6/76	
		Programmable Controller Reference and Application Manual	30000-90066	6.00	6/76	10/76
2/75	7/76	Programmable Controller Reference and Application Manual	30300-90002	12.00		
		QUERY Reference Manual	30000-90042	6.50	6/76	
3/76		QUERY/3000 Reference Manual	32216-90001	7.00		
		Real-Time Programmable Controller Reference and Application Manual	30000-90067	7.50	6/76	
2/75	7/76	Real-Time Programmable Controller Reference and Application Manual	30301-90002	9.50		
		Scientific Library Reference Manual	30000-90027	5.00	6/76	
7/75		Scientific Library Reference Manual, HP 3000	03000-90010	7.00		
		Site Preparation Manual	30000-90016	6.00	6/76	10/76
		Site Planning Workbook, HP Computer System	30000-90017	10.00	6/76	
8/76		Sort/3000 Reference Manual	32214-90001	6.50	8/76	
7/75		Student Assignment System Reference Manual	32901-90001	10.00	7/75	8/76
7/75		Student Assignment System — Technical Manual	32901-90005	13.00	7/75	
9/74		Student Information System Reference Manual	32900-90001	18.00	9/74	8/76
9/74		Student Information System — System Overview	32900-90002	7.00	9/74	
3/75		Student Information System -- Technical Manual	32900-90005	18.50	3/75	
		Systems Reference Manual	30000-90020	9.50	6/76	9/76
9/73		Systems Reference Manual, HP 3000 Computer	03000-90019	14.00		
6/76		Trace/3000 Reference Manual	03000-90015	7.00	6/76	

training schedule

The schedule for customer training courses on General Systems Division products is outlined below, and in the 2000 section of this publication. Included here are HP 3000 software courses offered in the U.S. and in Europe for the period November 1976 through January 1977. You can also obtain a copy of the schedule from your local HP sales office. A European course schedule is available through the sales offices in Europe; a U.S. schedule through U.S. sales offices.

Registration

Requests for enrollment in any of the courses should be made through your local HP Sales Office. Your Sales Representative will supply the Training Registrar at the appropriate location with the course number, dates, and requested motel reservations. Enrollments are acknowledged by a written confirmation indicating the training course, time of class, location and accommodations reserved.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE (U.S.)

November 1976 - February 1977

				Course Dates and Training Center Location	
NUMBER	COURSE TITLE	LENGTH	PRICE	WESTERN TECHNICAL CENTER	EASTERN TECHNICAL CENTER
22801A	3000 Series II, A Comprehensive Introduction	5 days	$500	12/13/76 1/03/76 1/17/77 1/31/77 2/14/77 2/28/77	11/01/76 12/13/76 1/03/77 1/24/77 2/07/77 2/28/77
22802A	3000 Series II, System Management and Operation	4 days	400	12/06/76 12/20/76 1/10/77 1/24/77 2/7/77 2/21/77	11/08/76 12/20/76 1/10/77 1/31/77 2/14/77
22956A	3000 IMAGE	5 days	500	12/13/76 1/17/77 1/31/77 2/28/77	11/15/76 1/17/77
22957A	3000 COBOL, Audio Self Study	30 hrs.	325	These courses can be ordered using the Direct Mail Order form in the back of the **Communicator.**	
22958A	3000 BASIC, Audio Self Study	30 hrs.	325		

Accommodations

Students provide their own transportation, meals, and lodging. The Training Registrar will be pleased to assist in securing motel reservations at the time your Sales Office requests a registration.

Cancellations

In the event you are unable to attend a class for which you are registered, please notify the Training Center Registrar immediately in order that we may offer your seat to another student. To avoid paying for a reservation which you do not use, we must receive notification of your cancellation no later than one week before the class begins.

GENERAL SYSTEMS DIVISION COURSE SCHEDULE (EUROPE)

November 1976 – January 1977

COURSE NUMBER	COURSE TITLE	LENGTH	FRANKFURT (GERMAN)	WINNERISH (ENGLISH)	ORSAY (FRENCH)	MILAN (ITALIAN)	STOCKHOLM (ENGLISH)
22801A	3000 Series II, A Comprehensive Introduction	5 days	11/08 1/10		1/10		11/08 12/06
22802A	3000 Series II, System Management and Operation	4 days	11/22 1/24		1/17	11/08	11/15 12/13
22956A	3000 IMAGE	5 days	12/06		11/15	12/13	

Technical Center Addresses

UNITED STATES

Eastern Technical Center
4 Choke Cherry Road
Rockville, Maryland 20850

Western Technical Center
5303 Stevens Creek Blvd.
Santa Clara, California 95050

EUROPE

Winnersh
King Street Lane
Winnersh, Wokingham
Berks RG11 5 AR
Tel: Wokingham 784774
Cable: Hewpie London
Tele: 847178 9

Orsay
Quartier de Courtaboeuf
Boite postale No. 6
F-91401-Orsay
France
Tel: (1) 907 78 25
Cable: HEWPACK Orsay
Telex: 60048

Frankfurt
Vertriebzentrale Frankfurt
Berner Strasse 117
Postale 560 140
D-6000 Frankfurt 56
Tel: 0611 5004
Telex: (841) 04-13249, 04-13081
Cable: HEWPACKSA Frankfurt

Milan
Via Amerigo Vespucci, 2
1-20124 Milan
Tel: (2) 62 51
Cable: HEWPACKIT Milano
Telex: 32046

Stockholm
Enighetsvagen 1-3, Fack
S-162 20 Bromma 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS
Stockholm Telex: 10721

subscription information

Annual subscriptions consisting of 6 issues are available as outlined below.

I. CUSTOMERS WITH SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS)

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract. A proration table appears on the ORDER FORM which is bound into this issue.

To receive a BASE SUBSCRIPTION at no charge as well as to purchase ADDITIONAL SUBSCRIPTIONS under the provisions of the Software Service Contract Program, complete the ORDER FORM and forward it to your local HP Sales and Service Office. Your local Customer Engineer will validate your order and mail it to the appropriate HP department.

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	NAC*	NAC*
ADDITIONAL SUBSCRIPTIONS (ea.)	$12/yr.	**

1) ADDITIONAL SUBSCRIPTIONS must go to the same name and address as the BASE SUBSCRIPTION to qualify for the reduced rates.

2) ADDITIONAL SUBSCRIPTIONS ordered at a later date than the BASE SUBSCRIPTION must include, with the order form, a copy of the address label for proper identification.

3) Charges for ADDITIONAL SUBSCRIPTIONS will be prorated to expire with your Software Service Contract.

4) Orders for ADDITIONAL SUBSCRIPTIONS from a customer with a Software Service Contract will be verified by the Customer Engineer who will complete the "FOR HP USE ONLY" portion of the subscription form and direct the order to the appropriate HP department. The customer will be billed by his local HP Customer Engineering Department.

*No Additional Charge (NAC)
**Contact your local HP Customer Engineer for the price in the currency of your country.

II. CUSTOMERS WITHOUT SOFTWARE MAINTENANCE AGREEMENTS OR SOFTWARE SUBSCRIPTION SERVICE AGREEMENTS (MAIL ORDER SUBSCRIPTIONS)

Rates:	U.S.A.	NON-U.S.A.
BASE SUBSCRIPTION	$48/yr.	***
ADDITIONAL SUBSCRIPTIONS (ea.)	$12/yr.	***

1) ADDITIONAL SUBSCRIPTIONS must be ordered at the same time as the BASE SUBSCRIPTION and go to the same name and address as the BASE SUBSCRIPTION to qualify for the reduced rate.

2) The customer is to include payment (check, bank draft, money order, etc.) with the order. This is a Direct Mail Order procedure; please do not send a purchase order to HP.

3) Complete the ORDER FORM as directed and mail together with your payment to:

Hewlett-Packard Co.
Mail Order Dept.
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A..

SUBSCRIPTION CORRESPONDENCE

Address all correspondence relating to **COMMUNICATOR** subscriptions to:

Subscription Service Manager
Hewlett-Packard Company
Mail Order Dept.
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A.

***The international customer is encouraged to also use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the order procedure outlines above. If the currency regulations in the customer's country disallow the purchase of bank drafts in American dollars, or if the customer does not have ready access to the required banking services, the customer may order subscriptions from the local HP Sales and Service Office through his Customer Engineer. The customer should contact his HP Office for the price of the subscription in the currency of his country then complete the ORDER FORM and forward it together with payment to his local HP Office.

HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ Title _____

Company _____

Street _____

City _____ State _____ Zip Code _____

Country _____

☐ MAIL ORDER SUBSCRIPTIONS

BASE SUBSCRIPTION $ _____

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT ENCLOSED $ _____

☐ SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS

BASE SUBSCRIPTION (NO ADDITIONAL CHARGE) NAC

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT YOU WILL BE BILLED $ _____

FOR HP USE ONLY

SUPPORT OFFICE NUMBER _____ ORDER DATE _____

APPROVED BY _____ C.E. NUMBER _____

SERVICE CONTRACT NUMBER _____ EXPIRATION DATE _____

AUTHORIZED TOTAL NUMBER OF SUBSCRIPTIONS _____

CUSTOMER'S HP OPERATING SYSTEM _____

Printed 4/76

TABLE OF PRORATED $ AMOUNT DUE PER ADDITIONAL SUBSCRIPTION

(Use only for ordering ADDITIONAL SUBSCRIPTION(S) against an existing Software Service Contract)

	Months Remaining in Service Contract											
	1	2	3	4	5	6	7	8	9	10	11	12
Cost of Each ADDITIONAL SUBSCRIPTION	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00

INSTRUCTIONS FOR ORDERING COMMUNICATOR

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract.

Customers who do not have Software Service Contracts may purchase Mail-Order Subscriptions through HP's Direct Mail Order System.

A. MAIL-ORDER SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due:
   a) Annual Base Subscription (6 issues) $ 48.00
   b) ____ Additional Subscriptions* @ $12.00 ea. $ ____
   c) Total Order Amount (a + b) $ ____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Mail check or bank draft with ORDER FORM to:

   HEWLETT-PACKARD CO.
   MAIL ORDER DEPARTMENT
   P.O. DRAWER #20
   MOUNTAIN VIEW, CA. 94043
   U.S.A.

B. SOFTWARE SERVICE CONTRACT SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due: (BASE SUBSCRIPTION is at no additional charge.)
   a) Annual Base Subscription (6 issues) $ 0.00
   b) ____ Additional Subscriptions* $ ____
   Prorate the dollar amount to make the ADDITIONAL SUBSCRIPTIONS EXPIRE WITH YOUR Software Service Contract. (SEE TABLE)
   c) Total Order Amount (a + b) $ ____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Forward ORDER FORM to your local HP Customer Engineering Representative. Your order will be approved and forwarded to the appropriate department. You will be billed for any ADDITIONAL SUBSCRIPTIONS by your local HP office.

C. SPECIAL INSTRUCTIONS FOR INTERNATIONAL CUSTOMERS

1. International customers who do not have Software Service Contracts are encouraged to use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the ordering procedures outlined in Instruction A above. Optionally, international customers may purchase the **Communicator** through their local HP Sales and Service Office. The customer should contact his HP Office for the subscription prices in the currency of his country, then complete the Order Form and forward it together with payment to his local HP Customer Engineering Department.

1. International customers with Software Service Contracts should follow the ordering procedure outlined in Instruction B above. If the customer wishes to purchase ADDITIONAL SUBSCRIPTIONS, he should contact the local HP Office for the subscription price in the currency of his country, then submit the ORDER FORM. The customer will be billed for ADDITIONAL SUBSCRIPTIONS by his local HP Office.

*All ADDITIONAL SUBSCRIPTIONS will be sent to the same name and address as the BASE SUBSCRIPTION.

47-5952-5570
Rev. 4/76

HEWLETT-PACKARD
COMPUTER SYSTEMS COMMUNICATOR ORDER FORM

Please Print:

Name _____ Title _____

Company _____

Street _____

City _____ State _____ Zip Code _____

Country _____

☐ MAIL ORDER SUBSCRIPTIONS

BASE SUBSCRIPTION $ _____

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT ENCLOSED $ _____

☐ SOFTWARE SERVICE CONTRACT SUBSCRIPTIONS

BASE SUBSCRIPTION (NO ADDITIONAL CHARGE) NAC

_____ ADDITIONAL SUBSCRIPTION(S) $ _____
No.

TOTAL AMOUNT YOU WILL BE BILLED $ _____

FOR HP USE ONLY

SUPPORT OFFICE NUMBER _____ ORDER DATE _____

APPROVED BY _____ C.E. NUMBER _____

SERVICE CONTRACT NUMBER _____ EXPIRATION DATE _____

AUTHORIZED TOTAL NUMBER OF SUBSCRIPTIONS _____

CUSTOMER'S HP OPERATING SYSTEM _____

Printed 4/76

TABLE OF PRORATED $ AMOUNT DUE PER ADDITIONAL SUBSCRIPTION

(Use only for ordering ADDITIONAL SUBSCRIPTION(S) against an existing Software Service Contract)

	Months Remaining in Service Contract											
	1	2	3	4	5	6	7	8	9	10	11	12
Cost of Each ADDITIONAL SUBSCRIPTION	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00	11.00	12.00

INSTRUCTIONS FOR ORDERING COMMUNICATOR

All Hewlett-Packard customers with Software Service Contracts are entitled to one BASE SUBSCRIPTION (1 copy per issue) at no additional charge. These customers may also buy ADDITIONAL SUBSCRIPTIONS whose purchase price is to be prorated against the remaining life of their Software Service Contract.

Customers who do not have Software Service Contracts may purchase Mail-Order Subscriptions through HP's Direct Mail Order System.

A. MAIL-ORDER SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due:
   a) Annual Base Subscription (6 issues) $ 48.00
   b) ____ Additional Subscriptions* @ $12.00 ea. $ ____
   c) Total Order Amount (a + b) $ ____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Mail check or bank draft with ORDER FORM to:

   HEWLETT-PACKARD CO.
   MAIL ORDER DEPARTMENT
   P.O. DRAWER #20
   MOUNTAIN VIEW, CA. 94043
   U.S.A.

B. SOFTWARE SERVICE CONTRACT SUBSCRIPTION(S)

1. Complete name and address portion of ORDER FORM.
2. Compute amount due: (BASE SUBSCRIPTION is at no additional charge.)
   a) Annual Base Subscription (6 issues) $ 0.00
   b) ____ Additional Subscriptions* $ ____
   Prorate the dollar amount to make the ADDITIONAL SUBSCRIPTIONS EXPIRE WITH YOUR Software Service Contract. (SEE TABLE)
   c) Total Order Amount (a + b) $ ____
   d) Transfer number of ADDITIONAL SUBSCRIPTIONS and all dollar amounts to ORDER FORM.
3. Forward ORDER FORM to your local HP Customer Engineering Representative. Your order will be approved and forwarded to the appropriate department. You will be billed for any ADDITIONAL SUBSCRIPTIONS by your local HP office.

C. SPECIAL INSTRUCTIONS FOR INTERNATIONAL CUSTOMERS

1. International customers who do not have Software Service Contracts are encouraged to use HP's Direct Mail Order System by remitting a bank draft in U.S. dollars according to the ordering procedures outlined in Instruction A above. Optionally, international customers may purchase the **Communicator** through their local HP Sales and Service Office. The customer should contact his HP Office for the subscription prices in the currency of his country, then complete the Order Form and forward it together with payment to his local HP Customer Engineering Department.

1. International customers with Software Service Contracts should follow the ordering procedure outlined in Instruction B above. If the customer wishes to purchase ADDITIONAL SUBSCRIPTIONS, he should contact the local HP Office for the subscription price in the currency of his country, then submit the ORDER FORM. The customer will be billed for ADDITIONAL SUBSCRIPTIONS by his local HP Office.

*All ADDITIONAL SUBSCRIPTIONS will be sent to the same name and address as the BASE SUBSCRIPTION.

47-5952-5570
Rev. 4/76

NOTE: This order form cannot be used for ordering manuals. It is for replacement pages only.

HEWLETT hp PACKARD

SOFTWARE/PUBLICATIONS DISTRIBUTION
ORDER FORM

UPDATES TO 3000 AND 2000
ACCESS LEVEL MANUALS ONLY

SHIP TO:

NAME _____

COMPANY _____

STREET _____

CITY _____ STATE _____ ZIP CODE _____

MANUAL NAME	PART NUMBER	QUANTITY

When completed, please mail this form to:

There is no charge for manual updates.

HEWLETT-PACKARD
SOFTWARE/PUBLICATIONS DISTRIBUTION
5303 Stevens Creek Blvd.
Santa Clara, CA 95050

NOT TO BE USED FOR ORDERING COMMUNICATOR SUBSCRIPTIONS

HEWLETT hp PACKARD

CORPORATE PARTS CENTER

Direct Mail
Parts and Supplies Order Form

SHIP TO:

NAME _____

COMPANY _____ CUSTOMER REFERENCE # _____

STREET _____ TAXABLE*? _____

CITY _____ STATE _____ ZIP CODE _____

Item No.	Check Digit	Part No.	Qty.	Description	List Price Each	Extended Total

Special Instructions			
	Sub-total		
	Your State & Local Sales Taxes*		
	Handling Charge	1	50
	TOTAL		

*Tax is verified by computer according to your ZIP CODE. If no sales tax is added, your state exemption number must be provided: #_____. If not, your order may have to be returned.

Check or Money Order, made payable to Hewlett-Packard Company, must accompany order.

When completed, please mail this form with payment to:

HEWLETT-PACKARD COMPANY
Mail Order Department
P.O. Drawer #20
Mountain View, CA 94043

Phone: (415) 968-9200

Most orders are shipped within 24 hours of receipt. Shipments to California, Oregon and Washington will be made via UPS. Other shipments will be sent Air Parcel Post, with the exception that shipments over 25 pounds will be made via truck. No Direct Mail Order can be shipped outside the U.S.

Although every effort is made to insure the accuracy of the data presented in the **Communicator**, Hewlett-Packard cannot assume liability for the information contained herein.

Prices quoted apply only in U.S.A. If outside the U.S., contact your local sales and service office for prices in your country.

direct mail order form

To expedite your order for software, manuals, or other materials described in this publication, use this form to order directly from the Corporate Parts Center in Mountain View, California.

1. Enter your name, address, customer reference number, and tax exemption information.
2. List the item or items you want by part number and description.
3. Compute the amount due and enclose a check or money order payable to Hewlett-Packard.

If you need assistance in placing your order, contact your local HP Sales Office.

Computer Systems **Communicator**
Subscription Service Manager
Hewlett-Packard Company
Mail Order Department
P.O. Drawer No. 20
Mountain View, California 94043
U.S.A. November/December 1976

Bulk Rate
U.S. Postage
PAID
Sunnyvale, CA. 94086
Permit No. 541

Address Correction Requested
Forwarding and Return Postage Guaranteed